平成 27 年度　修士論文

Group Shuffled BP復号法の新しいグループ分割手法について

電気通信大学大学院 情報システム学研究科
情報ネットワークシステム学専攻

学籍番号：1452024
吉田 恭平

指導教員
森田 啓義 教授
小川 朋宏 准教授
山口 和彦 准教授

平成 28 年 1 月 28 日

目 次

第1章　序論

1.1　研究背景と目的

近年，誤り訂正符号としてGallagerによって1963年に提案された低密度パリティ検査(LDPC)符号[1]が注目されている．LDPC符号の検査行列は，タナーグラフと呼ばれる２部グラフで表すことが可能であり，LDPC符号の代表的な復号法であるBelief Propagation(BP)復号法[1, 2]（sum-product復号法とも呼ばれる）では，タナーグラフのノード間で確率の計算をメッセージ交換の形で行っている．LDPC符号とBP復号法の組み合せにより得られる復号特性は優れており，特に，長い符号長のLDPC符号を用いた場合に，Shannon限界に近い性能を示すことが知られている[3].

BP復号法では，各変数ノードが並列にメッセージの交換を行う．一方で，Group Shuffled BP(GSBP)復号法[9, 10]では，変数ノード集合を複数のグループに分割し，グループ内ではBP復号法と同じ並列処理を行い，グループ間では逐次的に，前のグループのメッセージ交換が終わり次第，次のグループのメッセージ交換を行う．この順で処理することにより，前のグループ内で更新したメッセージを以降のグループで利用できるため，より信頼性の高いメッセージを得ることができ，結果としてより少ない反復回数で正確な復号が可能となる．

GSBP復号法では，受信語のビットを受信した順にグループに振り分けている．しかし，どのビットから更新を行うかで性能に差が生まれるため，次数（ノードからの辺の数）や誤りの伝播を考慮したグループ分割手法[11, 12]が提案されている．本論文では，既存のグループ分割手法とは異なる指標として，各変数ノードの局所的内径（その変数ノードを含む最短閉路の長さ）に注目してグループ分割を行うことを提案する．

BP復号法では，タナーグラフ上に短い閉路があると，確率計算が正確に行えず，性能が悪化することが知られている[13]．またGSBP復号法では，誤った計算が行われた場合，その結果も以降のグループに伝播してし

まう．そこで，局所的内径の大きい変数ノードからグループに振り分けて誤りの伝播を防ぐことで．どれほどの性能向上が見込めるかをシミュレーションを用いて評価する．特に本論文では，閉路の影響が顕著に出やすい符号長が短い符号を用いて，局所的内径の効果について確かめる．

1.2 研究の成果

計算機実験により，最大反復回数5回では，提案法は既存法に比べ少ないグループ数で優れた訂正能力を示した．例として，SNR= 4.5[dB] のグループ数が16の場合では，提案法によりビット誤り率が36%改善した．また，既存法と提案法の同じグループ数での平均反復回数を比較したところ，提案法により収束速度の向上が見られた．一方で，最大反復回数10回では，SNRの値が小さい場合は，提案法による訂正能力の改善が見られたが，SNRの値が高い場合では，提案法に比べ既存法の方が優れた訂正能力を示した．これは，提案法では，同じチェックノードと隣接する変数ノードが同じグループの振り分けられやすくなるため，更新メッセージの利用回数が既存法より少なくなるためと考えられる．

1.3 本論文の構成

論文の構成は以下の通りである．2章では，LDPC符号やそのタナーグラフについてなど，論文構成の上での諸準備を行う．3章では，LDPC符号の検査行列の構成法と，既存の復号法であるBP復号法とGSBP復号法について述べる．4章では，局所的内径を元にしたグループ分割手法を提案し，5章でその性能評価を行う．6章では，本論文のまとめと今後の課題について述べる．

第2章　準備

本章では，本論文で用いる基礎的な知識について述べる．まず2.1節で，２元線形符号とLDPC符号，IRA符号について述べる．次に2.2節で，グラフ理論を説明した後，検査行列から定義されるタナーグラフと，タナーグラフに関するパラメータとして局所的内径について述べる．最後に，2.3節で，本研究で想定する通信モデルとAWGN通信路を説明する．

2.1　線形符号

2.1.1　２元線形符号

あるアルファベット（シンボルの有限集合）$\mathcal{Q}$と正の整数Nが与えられたとき，符号長がNの**符号**$\mathcal{C}$とは，$\mathcal{Q}^N$の部分集合である．また，$\mathcal{C}$の要素$\boldsymbol{b}$を**符号語**と呼ぶ．特に$\mathcal{Q}=\mathbb{F}_2=\{0,1\}$，かつ$\mathcal{C}$が$\mathbb{F}_2^N$の線形部分空間であるとき，$\mathcal{C}$を**２元線形符号**とよぶ．本論文では，符号$\mathcal{C}$は常に符号長が$N$の２元線形符号とし，計算は全て2を法として行うものとする．

符号$\mathcal{C}$の次元を$K(\leq N)$とすると，$\mathcal{C}$はK個の一次独立なベクトルの組$(\boldsymbol{g}_1,\boldsymbol{g}_2,...,\boldsymbol{g}_K)$ $(\boldsymbol{g}_i\in\mathbb{F}_2^N,\ i=1,2,\ldots,K)$により張られる．すなわち，$\mathcal{C}$は

$$\mathcal{C}=\{\boldsymbol{b}\in\mathbb{F}_2^N:\boldsymbol{b}=a_1\boldsymbol{g}_1+a_2\boldsymbol{g}_2+\cdots+a_K\boldsymbol{g}_K,\ a_i\in\mathbb{F}_2,\ i=1,2,\ldots,K\} \tag{2.1}$$

で表現できる．また，次元を符号長で割った値$R=K/N$を符号$\mathcal{C}$の**符号化率**と呼ぶ．

上記$\boldsymbol{g}_i$を第i行に持つ$K\times N$行列$\boldsymbol{G}\in\mathbb{F}_2^{K\times N}$

$$\boldsymbol{G}=\begin{pmatrix}\boldsymbol{g}_1\\ \boldsymbol{g}_2\\ \vdots\\ \boldsymbol{g}_K\end{pmatrix} \tag{2.2}$$

を$\mathcal{C}$の**生成行列**と呼ぶ．符号$\mathcal{C}$の各符号語$\boldsymbol{b}$は，生成行列$\boldsymbol{G}$とベクトル$\boldsymbol{a}=(a_1,a_2,\ldots,a_K)\in\mathbb{F}_2^K$を用いて，$\boldsymbol{b}=\boldsymbol{a}\boldsymbol{G}$と表現することができる．また，

符号$\mathcal{C}$の生成行列$\boldsymbol{G}$が与えられたとき，$\boldsymbol{G}\boldsymbol{H}^T = \boldsymbol{0}$を満たす$M(= N-K) \times N$行列$\boldsymbol{H}$を$\mathcal{C}$の**検査行列**と呼ぶ．ここで，$\boldsymbol{H}^T$は$\boldsymbol{H}$の転置行列である．したがって，符号$\mathcal{C}$は

$$\mathcal{C} = \{\boldsymbol{b} \in \mathbb{F}_2^N : \boldsymbol{b}\boldsymbol{H}^T = \boldsymbol{0}\} \tag{2.3}$$

とも定義できる．

2.1.2 LDPC符号・IRA符号

行列内の1の個数が非常に少ない行列を疎行列と呼ぶ．2元線形符号の検査行列$\boldsymbol{H} \in \mathbb{F}_2^{K \times N}$が疎行列となるとき，$\boldsymbol{H}$により表現される符号を2元**低密度パリティ検査（***Low Density Parity Check; LDPC***）符号**[1]と呼ぶ．

行列内の1の個数を**重み**と呼ぶ．行列の$\boldsymbol{H}$の各列の重みが全て等しく，かつ各行の重みが全て等しい場合，$\boldsymbol{H}$で表現されるLDPC符号を**正則***LDPC***符号**といい，それ以外の場合を**非正則***LDPC***符号**という．

特に$M \times N$の検査行列$\boldsymbol{H}$が，ランダムな$M \times (N-M)$疎行列$\boldsymbol{H}_u$と$M \times M$階段行列

$$\boldsymbol{H}_p = \begin{pmatrix} 1 & 0 & 0 & 0 & \ldots & 0 & 0 \\ 1 & 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 1 & 0 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \ldots & 1 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 1 & 1 \end{pmatrix}$$

を用いて

$$\boldsymbol{H} = [\boldsymbol{H}_u | \boldsymbol{H}_p]$$

と書けるとき，$\boldsymbol{H}$で表現されるLDPC符号を（組織）*Irregular Repeat Accumulate (IRA)* **符号**[4]と呼ぶ．IRA符号は，検査行列$\boldsymbol{H}$のみを用いて符号化と復号が容易にできるという特徴がある．IRA符号は，一般のLDPC符号と同等，もしくはそれ以上の誤り訂正能力を持つことが知られている[4].

2.2 グラフ理論

グラフ$\mathcal{G} = (\mathcal{V}, \mathcal{E})$は，頂点集合$\mathcal{V}$と辺集合$\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$で構成される．辺に向きをつけないとき（したがって$(u,v) \in \mathcal{E} \Leftrightarrow (v,u) \in \mathcal{E}$が成り立つとき）$\mathcal{G}$

を**無向グラフ**と呼び，辺に向きをつけるとき，$\mathcal{G}$を**有向グラフ**とよぶ．また，グラフ$\mathcal{G}=(\mathcal{V},\mathcal{E})$の頂点集合$\mathcal{V}$を

$$\text{各 } i=1,2 \text{ において } (u,v\in\mathcal{V}_i \Longrightarrow (u,v)\notin\mathcal{E})$$

を満たすように2つの頂点集合$\mathcal{V}_1,\mathcal{V}_2$に分割できるとき，$\mathcal{G}$を**2部グラフ**と呼び$\mathcal{G}=(\mathcal{V}_1\cup\mathcal{V}_2,\mathcal{E})$と表す．特に，$\mathcal{E}=\mathcal{V}_1\times\mathcal{V}_2$となるとき，$\mathcal{G}=(\mathcal{V}_1\cup\mathcal{V}_2,\mathcal{E})$を**完全2部グラフ**と呼ぶ．

2.2.1 タナーグラフ

LDPC符号の検査行列$\boldsymbol{H}$は，**タナーグラフ**と呼ばれる2部グラフで表現できる．$M\times N$の検査行列$\boldsymbol{H}$が与えられたとき，$\boldsymbol{H}$を表現するタナーグラフ$\mathcal{T}=(\mathcal{V}_1\cup\mathcal{V}_2,\mathcal{E})$には，$\boldsymbol{H}$の列に対応する変数ノード$v_i\in\mathcal{V}_1\,(i=1,2,\ldots,N)$と，$\boldsymbol{H}$の行に対応するチェックノード$c_j\in\mathcal{V}_2\,(j=1,2,\ldots,M)$の二種類のノードがあり，$H_{mn}=1\,(1\le n\le N, 1\le m\le M)$となる$v_n$と$c_m$を辺で結ぶことで構成される．図2.1に検査行列$\boldsymbol{H}$と，それを表現するタナーグラフの例を示す．

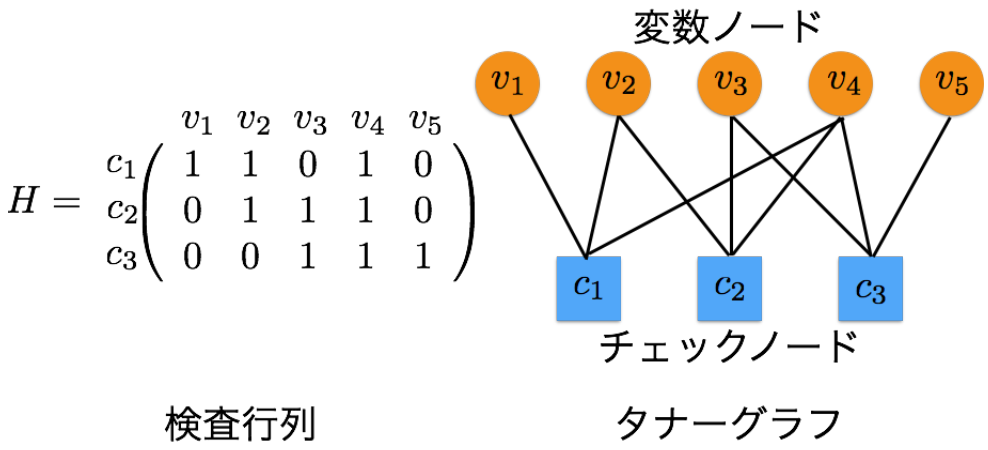

図 2.1: LDPC符号の検査行列$\boldsymbol{H}$と対応するタナーグラフ

タナーグラフのノードvから出ている辺の数をノードの**次数**と呼び，各ノードの次数を$d(v)\,(v\in\mathcal{V}_1\cup\mathcal{V}_2)$で表す．今，$d_v$を変数ノードの最大次数とする．このとき，変数ノードの次数分布多項式$\lambda(x)$は，変数ノードの次数の総数に対する次数がhとなる変数ノードの辺の総数の割合をλ_h,

つまり

$$\lambda_h = \frac{\sum_{\hat{n}:d(v_{\hat{n}})=h} d(v_{\hat{n}})}{\sum_{n=1}^{N} d(v_n)} = \frac{h \times |\{v_{\hat{n}} : d(v_{\hat{n}}) = h\}|}{\sum_{n=1}^{N} d(v_n)} \tag{2.4}$$

を用いて

$$\lambda(x) = \sum_{h=1}^{d_v} \lambda_h x^{h-1} \tag{2.5}$$

で定義される．同様に，d_cをチェックノードの最大次数とし，チェックノードの次数の総数に対する次数hとなるチェックノードの辺の総数の割合をρ_h，つまり

$$\rho_h = \frac{\sum_{\hat{m}:d(c_{\hat{m}})=h} d(c_{\hat{m}})}{\sum_{m=1}^{M} d(c_m)} = \frac{h \times |\{c_{\hat{m}} : d(c_{\hat{m}}) = h\}|}{\sum_{m=1}^{M} d(c_m)} \tag{2.6}$$

とすると，チェックノードの次数分布多項式$\rho(x)$は

$$\rho(x) = \sum_{h=1}^{d_c} \rho_h x^{h-1} \tag{2.7}$$

で定義される．

タナーグラフの変数ノードに符号語のビットを対応させると，タナーグラフは符号の制約条件を示している．つまり，どのチェックノードc_jにおいても，c_jに隣接する変数ノードに対応するビットの総和は0になることと，変数ノードに符号語のビットが対応していることは同値である．

例 1 図*2.1*において，符号語$\boldsymbol{b} = (b_1, b_2, b_3, b_4, b_5)$を変数ノード$v_1, v_2, v_3, v_4, v_5$にそれぞれ対応させる．このとき，次の式が成り立つ．

$$\begin{aligned} b_1 + b_2 + b_4 &= 0 \\ b_2 + b_3 + b_4 &= 0 \\ b_3 + b_4 + b_5 &= 0 \end{aligned} \tag{2.8}$$

2.2.2 局所的内径

一般のグラフ$\mathcal{G} = (\mathcal{V}, \mathcal{E})$において，**道**$\pi : u_0, u_1, \ldots, u_k$とは

1. $(u_i, u_{i+1}) \in \mathcal{E}\ (0 \leq i \leq k-1)$; かつ
2. $(u_i, u_{i+1}) = (u_{i'}, u_{i'+1})$となる整数組$(i, i')$（ただし$i \neq i'$）は存在しない．

を満たすノードの列であり，その長さ（遷移回数k）を$\ell(\pi)$で表す．特に$u_0 = u_k$となるときを**閉路**と呼ぶ．

今，ノードuを含む閉路の集まりを$\mathcal{C}(u)$とすると，uの**局所的内径**$g(u)$とは，uを含む最短閉路の長さ，つまり

$$g(u) = \min_{p \in \mathcal{C}(u)} \ell(\pi) \tag{2.9}$$

で定義される．ただし，どの閉路にも含まれないノードの局所的内径は任意の大きな値とする．

例 2 図 *2.2*のタナーグラフを考える．変数ノードv_1とv_2においては，閉路$\pi : v_1, c_1, v_2, c_2, v_1$が$v_1$や$v_2$を含む最短閉路となるので，$g(v_1) = g(v_2) = 4$である．同様に，$v_3$と$v_4$においては，閉路$\mu : v_3, c_3, v_4, c_2, v_1, c_1, v_3$が$v_3$や$v_4$を含む最短閉路となるので，$g(v_3) = g(v_4) = 6$である．一方，$v_5$や$v_6$を含む閉路は存在しないので，$g(v_5)$と$g(v_6)$は任意の大きな値（例えば100など）とする．

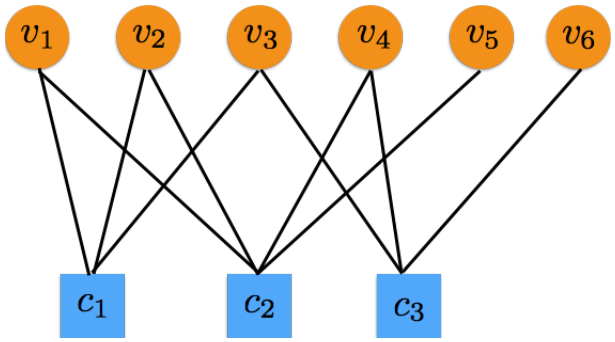

図 2.2: タナーグラフとノードの局所的内径

2.3 通信路

2.3.1 通信モデル

本論文では，図2.3に示す通信モデルを考える．

いま，送信者が情報$\boldsymbol{a} = (a_1, a_2, \ldots, a_K) \in \mathbb{F}_2^N$を送信したいとする．まず送信者は，情報$\boldsymbol{a}$を生成行列$\boldsymbol{G}$を用いて，長さ$N$ $(N > K)$の符号語$\boldsymbol{b} = \boldsymbol{aG} = (b_1, b_2, ..., b_N) \in \mathbb{F}_2^N$に変換する．その後，符号語$\boldsymbol{b}$をBPSK変調により電気信号に変換し，通信路を通して受信者に送る．受信者側では，

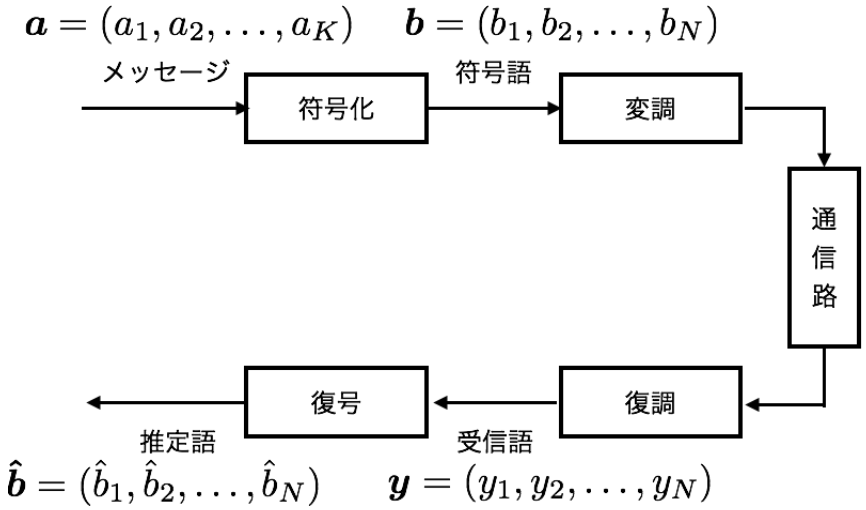

図 2.3: 通信モデル

受け取ったを信号を復調し受信語$\boldsymbol{y} = (y_1, y_2, \ldots, y_N) \in \mathbb{R}^N$を得る．その後，復号により，符号語の推定語$\hat{\boldsymbol{b}} = (\hat{b}_1, \hat{b}_2, \ldots, \hat{b}_N) \in \mathbb{F}_2^N$を得る．

2.3.2 AWGN通信路

本論文では，2値入力のAWGN通信路を仮定する．はじめに，式(2.10)を用いて符号語$\boldsymbol{b} = (b_1, b_2, ..., b_N) \in \mathbb{F}_2^N$をBPSK変調により$\boldsymbol{s} = (s_1, s_2, ..., s_N) \in \{-1, 1\}^N$に変換する．

$$s_i = \begin{cases} 1 & (b_i = 0) \\ -1 & (b_i = 1) \end{cases}, \ (i = 1, 2, \ldots, N) \tag{2.10}$$

得られた$\boldsymbol{s}$がAWGN通信路に入力され，AWGN通信路内で，白色ガウス雑音$\boldsymbol{z} = (z_1, z_2, ..., z_N) \in \mathbb{R}^N$が付加されたものが受信語$\boldsymbol{y} = (y_1, y_2, \ldots, y_N) \in \mathbb{R}^N$となる．すなわち，$i$ビット目の受信語のシンボル$y_i$は，

$$y_i = s_i + z_i, \ i = 1, 2, \ldots, N \tag{2.11}$$

となる．ここで，z_iは平均0，分散σ^2の正規分布の確率密度関数

$$P(z_i) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{{z_i}^2}{2\sigma^2}\right), \ i = 1, 2, \ldots, N \tag{2.12}$$

に従う白色ガウス雑音とする．

また，E_bを情報1ビット当たりのエネルギー，N_0を片側電力エネルギーとし，AWGN通信路の信号対雑音比(Signal to Noise ratio : SNR)E_b/N_0を

$$\frac{E_b}{N_0} \triangleq \frac{1}{2\sigma^2 R} \tag{2.13}$$

と定義する．SNRの単位は[dB]となり，SNR $E_b/N_0 = A$[dB]のとき，$\sigma^2 = 10^{-(\frac{A}{10})}/2R$となる．SNR E_b/N_0は，通信路を評価する際の指標であり，SNRが低いほど雑音の影響を受けやすく，SNRが高いほど雑音の影響を受けにくくなる．

第3章　検査行列の構成法と Belief Propagation復号法

本章では，LDPC符号の検査行列$\boldsymbol{H}$の構成法とBelief Propagation(BP)復号法[1]について述べる．まず3.1節で，LDPC符号の検査行列$\boldsymbol{H}$の構成法として，Gallagerにより提案された構成法[1]と，立川により提案された構成法[7]について述べる．3.2節では，LDPC符号の復号法であるBP復号法[1]を説明した後，その改良手法であるShuffled BP(SBP)復号法[9, 10], Group Shuffled BP(GSBP)復号法[9, 10]について述べる．

3.1　LDPC符号の構成法

3.1.1　Gallagerの構成法

1963年，GallagerによりLDPC符号の検査行列$\boldsymbol{H}$の構成法が提案された[1]．Gallagerの構成法では，列の重みが1となる部分行列と，部分行列にランダムな列置換を行った行列を結合することで検査行列を構成する．

いま，行の重みがw_r，列の重みがw_cとなる$M \times N$の正則LDPC符号の検査行列を構成することを考える．

Gallagerによる構成法は次の通りである．まず，行の重みが全てw_r，列の重みが全て1となる$\frac{N}{w_r} \times N$行列$\boldsymbol{H}'_1$を構成する．次に，行列$\boldsymbol{H}'_1$にランダムな列置換を行い，$w_c - 1$個の行列$\boldsymbol{H}'_2, \boldsymbol{H}'_3, \ldots, \boldsymbol{H}'_{w_c}$を生成する．最後に，$\boldsymbol{H}'_1, \boldsymbol{H}'_2, \ldots, \boldsymbol{H}'_{w_c}$を結合することで行の重みが$w_r$，列の重みが$w_c$の$M \times N$（ただし，$M = \frac{N}{w_r} \times w_c$）の正則LDPC符号の検査行列を得る．Gallagerの構成法では，$\frac{N}{w_r}$が整数である場合，かつ正則LDPC符号の場合にのみ構成が可能である．

例 3 *Gallager*の構成法により構成した行の重みが4，列の重みが3となる

9×12検査行列の例を次に示す.

$$\left(\begin{array}{cccccccccccc}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\
\hline
0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\
\hline
1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0
\end{array}\right) \tag{3.1}$$

行列*(3.1)*の黒線で区切られた各部分行列は，行の重みが4，列の重みが1の3×12行列となっている.

3.1.2 立川の構成法

Gallagerの構成法では，$\frac{N}{w_r}$が整数である場合，かつ正則LDPC符号の場合にのみ構成が可能という制約があった．しかし，実際には，通信環境等の関係で，任意の重みの非正則なLDPC符号の検査行列が必要となる場合がある．そこで，立川により，最大フローアルゴリズムを利用した，任意の重みの検査行列を構成する手法[7]が提案されている．本小節では，最大フローアルゴリズムと，立川の構成法について述べる.

最大フローアルゴリズム

最大フローアルゴリズム（詳細は文献[6]参照）は，各辺に容量(capacity)が設定されたネットワークで，フロー(flow)と呼ばれるデータ流量の最大値を求めるアルゴリズムである.

入次数が0となる始点をs，出次数が0となる終点をtとし，sとtを固定した有向グラフ$\mathcal{G} = (\mathcal{V}, \mathcal{E})$を考える．ノード$v$の入辺集合を$In(v) = \{e \in \mathcal{E} | \exists u \in \mathcal{V} s.t.\ e = (u, v)\}$，出辺集合を$Out(v) = \{e \in \mathcal{E} | \exists u \in \mathcal{V} s.t.\ e = (v, u)\}$とし，辺集合$\mathcal{E}$に対して，容量$cap_{\mathcal{G}} : \mathcal{E} \to \mathbb{R}^+$を定義する．また，容量が与えられたとき，フローを次の1), 2)を満たす関数$flow_{\mathcal{G}} : \mathcal{E} \to \mathbb{R}^+$と定義する.

1) 各辺$e \in \mathcal{E}$に対して，$flow_{\mathcal{G}}(e) \leq cap_{\mathcal{G}}(e)$が成り立つ.

2) 各ノード$v \in \mathcal{V} \backslash \{s, t\}$に対して，$\sum_{e \in In(v)} flow_{\mathcal{G}}(e) = \sum_{e \in Out(v)} flow_{\mathcal{G}}(e)$が成り立つ.

有向グラフ$\mathcal{G}$，容量 $cap_{\mathcal{G}}$，始点s，終点tから構成される４つ組をネットワーク$\mathcal{N} = (\mathcal{G}, s, t, cap_{\mathcal{G}})$と呼ぶ．

最大フローアルゴリズムでは，以下に定義する**残余ネットワーク**$\mathcal{N}_f$を用いる．$\mathcal{G}$の逆向きの有向辺集合を$\mathcal{E}_{\mathcal{R}}$としたとき，有向グラフ$\mathcal{G}'$の辺集合$\mathcal{E}_{\psi}$と容量 cap_{ψ}を次のように定義する．

- $\mathcal{E}_{\psi} = \{e \in \mathcal{E} | flow_{\mathcal{G}}(e) < cap_{\mathcal{G}}(e)\} \cup \{e_{\mathcal{R}} \in \mathcal{E}_{\mathcal{R}} | flow_{\mathcal{G}}(e) > 0\}$
- $flow_{\mathcal{G}}(e) < cap_{\mathcal{G}}(e)$のとき，$cap_{\psi}(e) = cap_{\mathcal{G}}(e) - flow_{\mathcal{G}}(e)$
- $flow_{\mathcal{G}}(e) > 0$のとき，$cap_{\psi}(e_{\mathcal{R}}) = flow_{\mathcal{G}}(e)$

このとき，有向グラフ$\mathcal{G}' = (\mathcal{V}, \mathcal{E}_{\psi})$，始点$s$，終点$t$，容量 cap_{ψ}から構成される４つ組を残余ネットワーク$\mathcal{N}_f = (\mathcal{G}', s, t, cap_{\psi})$と定義する．

最大フローアルゴリズムの入力はネットワーク$\mathcal{N}$であり，出力は

$$\max \sum flow_{\mathcal{G}}(\hat{e}) \qquad \text{（最大フロー）} \tag{3.2}$$

である．ここで，出力の式の$\sum$は，始点sからの出辺$\hat{e}$のフローの和を意味する．

以下に最大フローアルゴリズムを示す.

最大フローアルゴリズム

初期設定 全ての辺$e \in \mathcal{E}$について$flow_{\mathcal{G}}(e) = 0$とする．

ステップ１ ネットワーク$\mathcal{N}$の残余ネットワーク$\mathcal{N}_f = (\mathcal{G}', s, t, cap_{\psi})$を構成する．

ステップ２ $\mathcal{N}_f$の始点sから終点tまでの道πを幅優先探索で見つける．道πが存在しない場合，ステップ４へ進む．

ステップ３ 道$\pi = u_1, u_2, \ldots, u_{j-1}, u_j$が存在するとき，$\Delta(\pi)$を道$\pi$に含まれる辺の容量の最小値，つまり

$$\Delta(\pi) = \min_{1 \leq i \leq j-1} (cap_{\psi}((u_i, u_{i+1}))) \tag{3.3}$$

とし，$(u_i, u_i + 1)$のフローを次のように更新する．

- $(u_i, u_i + 1) \in \mathcal{E}$の場合：$flow_{\mathcal{G}}((u_i, u_i + 1)) := flow_{\mathcal{G}}((u_i, u_i + 1)) + \Delta(\pi)$

- $(u_i, u_i+1) \in \mathcal{E}_{\mathcal{R}}$ の場合：$flow_{\mathcal{G}}((u_i, u_i+1)) := flow_{\mathcal{G}}((u_i, u_i+1)) - \Delta(\pi)$

フローを更新した後，ステップ 1 へ戻り残余ネットワークを更新し，ステップ 2 へ進む．

ステップ 4 始点 s から出ている辺 $\hat{e}$ のフローの総和を最大フローとして出力し終了．

立川の構成法のアルゴリズム

検査行列はタナーグラフにより定義することが可能であるため，検査行列を構成する場合はタナーグラフを求めれば良い．つまり，所望の $M \times N$ の検査行列を構成することは，濃度が $|\mathcal{V}_1| = N$，$|\mathcal{V}_2| = M$ となる完全 2 部グラフ $W = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{V}_1 \times \mathcal{V}_2)$ を考え，行重み，列重みに対応する次数の条件を満たす W の部分グラフをタナーグラフとして求めることと同義である．立川の構成法では，有向完全 2 部グラフ B に，始点 s と終点 t，およびそれらに付随する有向辺を加えたグラフ $\mathcal{G}$ を用意し，$\mathcal{G}$ の各辺に容量を定義したネットワーク $\mathcal{N} = (\mathcal{G}, s, t, cap_{\mathcal{G}})$ の最大フローを求めることで所望のタナーグラフを生成している．

いま，所望するタナーグラフ $\mathcal{T} = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E})$ の $\mathcal{V}_1$ の変数ノード $v_1, v_2, \ldots, v_N$ の次数をそれぞれ $\phi'_1, \phi'_2, \ldots, \phi'_N$ と，$\mathcal{V}_2$ のチェックノード $c_1, c_2, \ldots, c_M$ の次数をそれぞれ $\omega'_1, \omega'_2, \ldots, \omega'_M$ とする．以下に，立川の構成法のアルゴリズムを示す．

入力

- 変数ノードの個数 N
- チェックノードの個数 M
- 各変数ノードの次数 $\phi'_1, \phi'_2, \ldots, \phi'_N$
- 各チェックノードの次数 $\omega'_1, \omega'_2, \ldots, \omega'_M$

出力

- 変数ノードの個数 N，チェックノードの個数 M，各変数ノードの次数が $\phi'_1, \phi'_2, \ldots, \phi'_N$，各チェックノードの次数が $\omega'_1, \omega'_2, \ldots, \omega'_M$ となるタナーグラフ

立川の構成法のアルゴリズム

ステップ１ $|\mathcal{V}_1| = N$，$|\mathcal{V}_2| = M$，かつ，$\mathcal{V}_1$から$\mathcal{V}_2$への有向辺を持つ有向完全２部グラフ$B = (\mathcal{V}_1 \cup \mathcal{V}_2, \mathcal{E})$を構成する．

ステップ２ 始点sと終点t追加し，sから$\mathcal{V}_1$の各ノードに有向辺をひき，$\mathcal{V}_2$の各ノードからtへ有向辺をひく．Bにsとtおよびそれらに付随する有向辺を追加したグラフを$\mathcal{G}$とする．また，各$v_n \in \mathcal{V}_1$に対して，容量 $cap_{\mathcal{G}}((s, v_n))$を

$$cap_{\mathcal{G}}((s, v_n)) = \phi'_n \tag{3.4}$$

とする．同様に，各$c_m \in \mathcal{V}_2$に対して，容量 $cap_{\mathcal{G}}((c_m, t))$を

$$cap_{\mathcal{G}}((c_m, t)) = \omega'_m \tag{3.5}$$

とする．また，もともとBに含まれていた辺eについては

$$cap_{\mathcal{G}}(e) = 1 \tag{3.6}$$

とする．そして，グラフ$\mathcal{G}$，始点s，終点t，容量$cap_{\mathcal{G}}$の４つ組からなるネットワーク$\mathcal{N} = (\mathcal{G}, s, t, cap_{\mathcal{G}})$を構成する．

ステップ３ ネットワーク$\mathcal{N} = (\mathcal{G}, s, t, cap_{\mathcal{G}})$に対して最大フローアルゴリズムを適用する．ただし，幅優先探索時にノードのindexをランダムに探索するものとする．つまり，すべての道を探す場合で，indexを入れ替える．道が見つからない場合は，アルゴリズムを終了する．

ステップ４ 最大フローアルゴリズムで出力された最大フローが$\sum_{n=1}^{N} \phi'_n$と一致するか確認する．もし，一致しない場合，タナーグラフは存在しないと返す．一致する場合は，フローが0となる全ての辺および始点sとsに繋がる辺，終点tとtに繋がる辺を削除し，残ったグラフのベースグラフ（有向辺を無向辺として得られる無向グラフ）を所望タナーグラフとして出力する．

例 4 立川の構成法を用いて，変数ノードの個数$N = 5$，チェックノードの個数$M = 3$，各変数ノードの次数が$\phi'_1 = \phi'_2 = 2, \phi'_3 = \phi'_4 = \phi'_5 = 1$，各チェックノードの次数が$\omega'_1 = \omega'_2 = 2, \omega'_3 = 3$となるタナーグラフを構成することを考える．

まず，ステップ１，ステップ２により図*3.1*に示すネットワーク$\mathcal{N}$を構成する．ここで，図中の赤字は辺の容量を表す.

次に，ステップ３において，ネットワーク$\mathcal{N}$に対して最大フローアルゴリズムを適用して，最大フローを達成する道を表す図*3.2*と最大フローを得る．ここで，図中の黒い辺はフローが0の辺，赤い辺がフローが非0の辺で赤い数字がフローの値を表す.

最後に，ステップ4では，最大フローと変数ノードの次数の和が一致するので，フローが0となる全ての辺および始点sとsに繋がる辺，終点tとtに繋がる辺を削除し，残ったグラフの有向辺を無向辺としたグラフをタナーグラフとして出力する．

立川の構成法を用いることで，任意の次数分布のLDPC符号の検査行列$\boldsymbol{H}$およびIRA符号の検査行列の部分行列$\boldsymbol{H}_u$を構成できる．本論文で使用する符号は，立川の構成法により構成される検査行列を用いる．

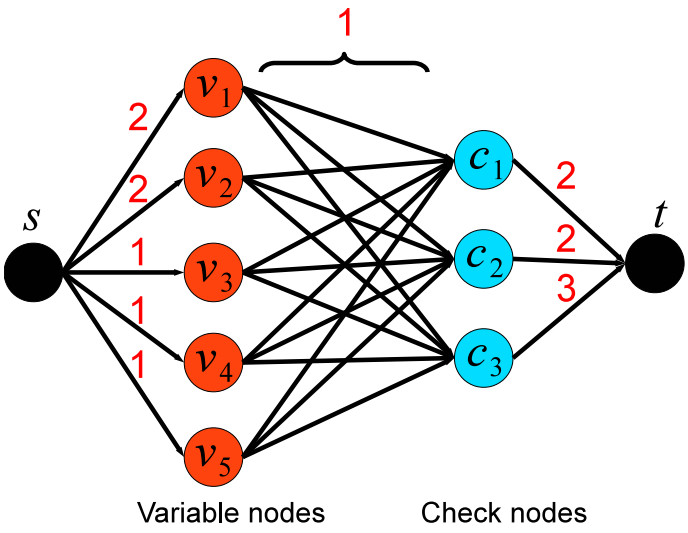

図 3.1: $N = 5$, $M = 3$ のネットワーク $\mathcal{N}$ の例（文献 [7] p.30 より引用）

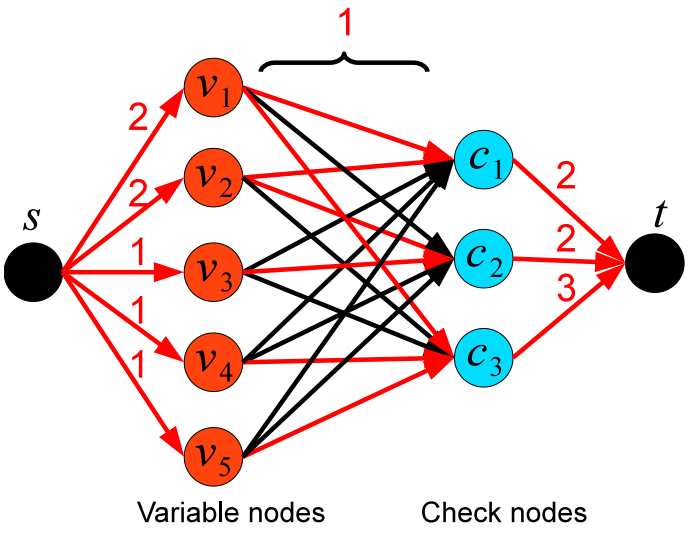

図 3.2: 最大フローアルゴリズムを適用したグラフの例（文献 [7] p.30 より引用）

3.2 LDPC符号のBP復号法

本節では，LDPC符号の代表的な復号法であるBelief Propagation（BP）復号法[1]と，その改良手法であるShuffled BP（SBP）復号法[9, 10]，Group Shuffled BP（GSBP）復号法[9, 10]について説明する．

チェックノードc_mと隣接する変数ノードの添字集合を$\mathcal{N}(m) = \{n : H_{mn} = 1\}$，変数ノード$v_n$と隣接するチェックノードの添字集合を$\mathcal{M}(n) = \{m : H_{mn} = 1\}$とする．また，符号語を$\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$，受信語を$\boldsymbol{y} = (y_1, y_2, \ldots, y_N)$とし，通信路はBPSK変調を伴うAWGN通信路を仮定する．

いま，送信した符号語のnビット目$(n = 1, 2, \ldots, N)$のシンボルが$x_n = 0$であるときに，受信シンボルが$y_n \in \{0, 1\}$ $(n = 1, 2, \ldots, N)$である確率を$P(y_n|x_n = 0)$，$x_n = 1$であるときに，受信シンボルが$y_n \in \{0, 1\}(n = 1, 2, \ldots, N)$である確率を$P(y_n|x_n = 1)$と表すことにする．受信シンボル$y_n(n = 1, 2, \ldots, N)$に対する対数尤度比を

$$L_n = \ln \frac{P(y_n|x_n = 0)}{P(y_n|x_n = 1)} \tag{3.7}$$

と定義する．また，各変数ノードは受信語の各ビットと対応しており，受信ビットの添字と変数ノードの添字は一致する．

3.2.1 BP復号法

BP復号法[1]は，ファクターグラフと呼ばれるグラフを元に，受信語$\boldsymbol{y}$に基づき事後確率が最大となる推定語$\hat{\boldsymbol{x}}$，つまり

$$\hat{\boldsymbol{x}} = \arg\max_{\boldsymbol{x} \in C} P(\boldsymbol{x}|\boldsymbol{y}) \tag{3.8}$$

を出力する復号法である．BP復号法では，各ビットの事後確率の周辺化計算を，タナーグラフ上でのメッセージ交換の形式で，効率よく計算する．各変数ノードが同時に変数ノード処理を，各チェックノードが同時にチェックノード処理を行い，その結果をメッセージの形で交換することを繰り返して復号する．反復l回目のチェックノードc_mから変数ノードv_n $(n \in \mathcal{N}(m))$へのメッセージを$\alpha_{mn}^{(l)}$，変数ノードv_nからチェックノードc_m $(m \in \mathcal{M}(n))$へのメッセージを$\beta_{mn}^{(l)}$とする．以下にBP復号法のアルゴリズムを示す（詳細は文献[8]などを参照）．

BP復号法のアルゴリズム

ステップ１（初期化） $H_{mn}=1$を満たす組(m,n)に対して$\beta_{mn}^{(0)}=L_n$とする．また，反復回数を表す変数lの初期値を1とし，最大反復回数をl_{max}とする．

ステップ２（メッセージ更新） $n=1,2,\ldots,N$について以下の二つの処理を行う．

行処理） $m\in\mathcal{M}(n)$において，$\alpha_{mn}^{(l)}$を次のように計算する．

$$\alpha_{mn}^{(l)}=2\tanh^{-1}\left(\prod_{n'\in\mathcal{N}(m)\backslash\{n\}}\tanh\left(\frac{\beta_{mn'}^{(l-1)}}{2}\right)\right) \tag{3.9}$$

ここで，$D\backslash\{d\}$は集合Dから要素dを除いた集合を表し，以降，簡単化のため$D\backslash d$と表す．

列処理） $m\in\mathcal{M}(n)$において，$\beta_{mn}^{(l)}$を次のように計算する．

$$\beta_{mn}^{(l)}=L_n+\sum_{m'\in\mathcal{M}(n)\backslash m}\alpha_{m'n}^{(l)} \tag{3.10}$$

ステップ３（一時推定ビットの決定） $n=1,2,\ldots,N$において推定ビットの決定に用いる値$\gamma_n^{(l)}$を次式で求める．

$$\gamma_n^{(l)}=L_n+\sum_{m'\in\mathcal{M}(n)}\alpha_{m'n}^{(l)} \tag{3.11}$$

反復l回目の推定ビット$\hat{\boldsymbol{x}}^{(l)}=(\hat{x}_1^{(l)},\hat{x}_2^{(l)},\ldots,\hat{x}_N^{(l)})$を次式によって決定する．

$$\hat{x}_n^{(l)}=\begin{cases}0 & \gamma_n^{(l)}\geq 0\\ 1 & \gamma_n^{(l)}<0\end{cases},n=0,1,\ldots,N \tag{3.12}$$

ステップ４（終了判定） $\hat{\boldsymbol{x}}^{(l)}\boldsymbol{H}^T=\boldsymbol{0}$の場合，または，反復回数$l$が$l_{max}$の場合は，$\hat{\boldsymbol{x}}^{(l)}=(\hat{x}_1^{(l)},\hat{x}_2^{(l)},\ldots,\hat{x}_N^{(l)})$を推定語として出力して終了する．そうでなければ，$l:=l+1$としてステップ２へ戻る．

ステップ２での行処理・列処理の図を図3.3に示す．また，行処理・列処理の反復の図を図3.4に示す．

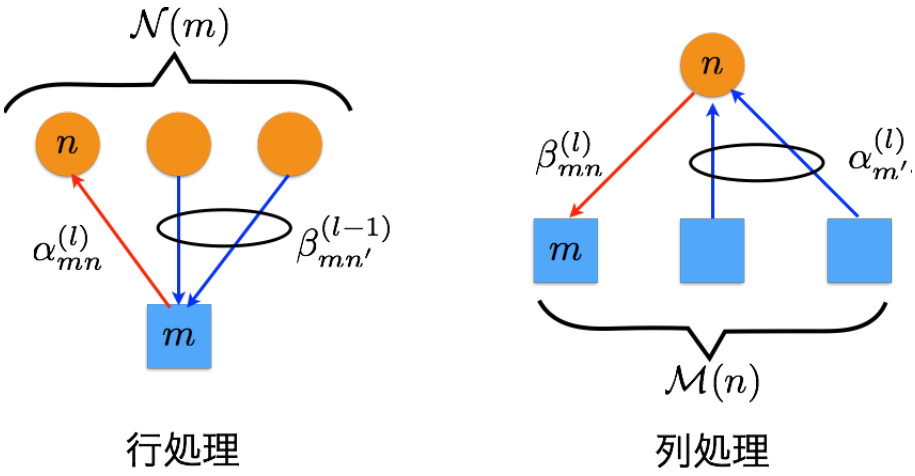

図 3.3: BP 復号法の行処理・列処理

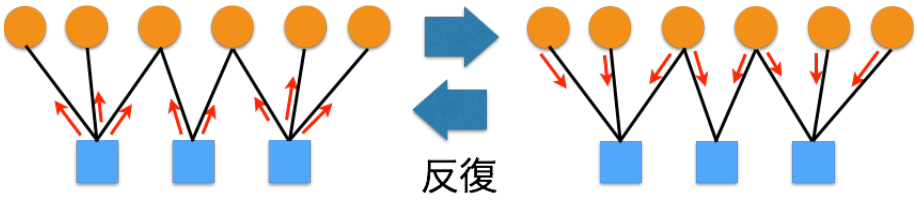

図 3.4: BP 復号法の行処理・列処理の反復

3.2.2 Shuffled BP 復号法

BP 復号法では，1 回の反復の処理で，全てのチェックノードでメッセージを同時に更新し，全ての変数ノードでメッセージを同時に更新している．そのため，各 $\alpha_{mn}^{(l)}$（反復 l 回目の c_m から v_n へのメッセージ）は，反復 $l-1$ 回の v_n 以外の c_m の隣接ノードから c_m に送られてくるメッセージ $\beta_{mn'}^{(l-1)}$ $(n' \in \mathcal{M}(m) \backslash n)$ すべてを用いて更新している．また，反復 l 回目の変数ノードからチェックノードへのメッセージ $\beta_{mn}^{(l)}$ は，$\alpha_{m'n}^{(l)}$ $(m' \in \mathcal{N}(m) \backslash m)$ を用いて更新している．しかし，メッセージは更新するたび信頼度が上がるため，反復 l 回目のメッセージ $\alpha_{mn}^{(l)}$ を更新する際には，反復 $l-1$ 回目のメッセージ $\beta_{mn'}^{(l-1)}$ よりも反復 l 回のメッセージ $\beta_{mn'}^{(l)}$ を用いて更新した方が，より信頼度の高い $\alpha_{mn}^{(l)}$ を得ることができる．

そこで，Shuffled BP（SBP）復号法 [9, 10] では，並列に行われていた処理

を，各ノードで逐次的に処理することで，可能な限り反復l回のメッセージ$\beta_{mn'}^{(l)}$を用いて更新する．これにより，より少ない反復回数で正確に計算することが期待できる．SBP復号法では，BP復号法のステップ２を以下のように変更する．

SBP復号法の行処理，列処理

ステップ２（メッセージ更新） $n = 1, 2, \ldots, N$に対して

行処理） $m \in \mathcal{M}(n)$に対して$\alpha_{mn}^{(l)}$を次のように計算する．

$$\alpha_{mn}^{(l)} = 2\tanh^{-1}\left(\prod_{\substack{n'\in\mathcal{N}(m)\backslash n,\\ n'<n}} \tanh\left(\frac{\beta_{mn'}^{(l)}}{2}\right) \times \prod_{\substack{n'\in\mathcal{N}(m)\backslash n,\\ n'>n}} \tanh\left(\frac{\beta_{mn'}^{(l-1)}}{2}\right)\right) \tag{3.13}$$

列処理） $m \in \mathcal{M}(n)$に対して$\beta_{mn}^{(l)}$を次のように計算する．

$$\beta_{mn}^{(l)} = L_n + \sum_{m'\in\mathcal{M}(n)\backslash m} \alpha_{m'n}^{(l)} \tag{3.14}$$

3.2.3 Group Shuffled BP復号法

SBP復号法では，各ノードで逐次処理することで，収束速度を向上させていた．しかし，逐次的な処理では，処理を並列に行うBP復号法に比べ，復号の際に遅延が発生してしまう問題点がある．そこで，Group Shuffled BP(GSBP)復号法[9, 10]では，ノードを複数のグループに分割し，グループ内では並列に，グループ間では逐次的に実行することで，遅延の減少を図っている（図3.5参照）．つまり，グループ数が1のGSBP復号法はBP復号法，グループ数がN（符号長）のGSBP復号法はSBP復号法となる．

ここでは，文献[9]を参考に，受信語の各添字集合G_kの濃度が（均等に）N_Gとなるように

$$G_i = \left\{ n \in \{1, 2, \ldots, N\} \;\middle|\; i = \left\lceil \frac{n}{N_G} \right\rceil \right\} \tag{3.15}$$

として，変数ノードの添字集合$\{1, 2, \ldots, N\}$を$r(= N/N_G)$個の集合（グループ）$G_1, G_2, \ldots, G_r$に分割する．GSBP復号法のアルゴリズムは，BP復号法のアルゴリズムのステップ２を以下のように変更する．

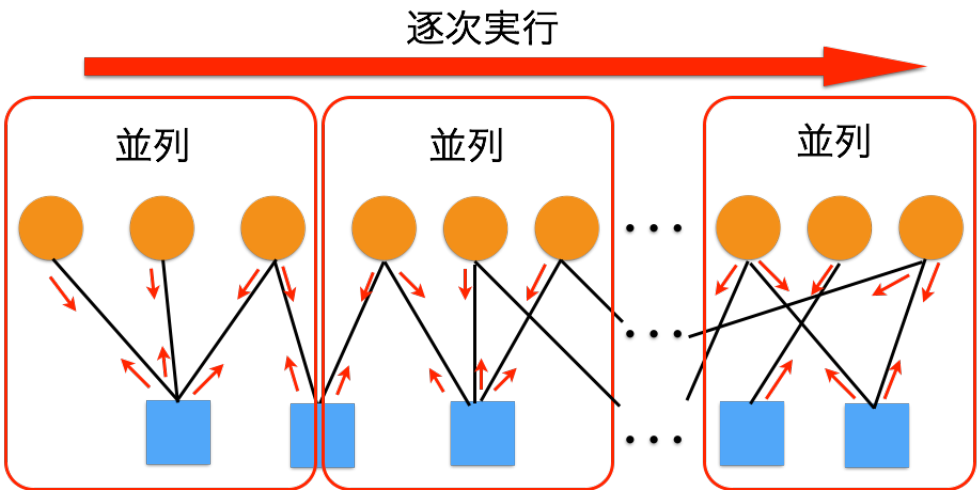

図 3.5: GSBP 復号法

GSBP 復号法の行処理，列処理

ステップ 2（メッセージ更新） $k = 1, 2, \ldots, r$ の順で，以下の 2 つの処理を行う

行処理） $n = (k-1)N_G + 1, (k-1)N_G + 2, \ldots, kN_G$ において $m \in \mathcal{M}(n)$ に対して $\alpha_{mn}^{(l)}$ を次のように計算する．

$$\alpha_{mn}^{(l)} = 2\tanh^{-1}\left(\prod_{\substack{n' \in \mathcal{N}(m)\backslash n, \\ n' \in G_{k'}(k'<k)}} \tanh\left(\frac{\beta_{mn'}^{(l)}}{2}\right) \times \prod_{\substack{n' \in \mathcal{N}(m)\backslash n, \\ n' \in G_{k'}(k' \geq k)}} \tanh\left(\frac{\beta_{mn'}^{(l-1)}}{2}\right)\right) \tag{3.16}$$

列処理） $n = (k-1)N_G + 1, (k-1)N_G + 2, \ldots, kN_G$ において $m \in \mathcal{M}(n)$ に対して $\beta_{mn}^{(l)}$ を次のように計算する．

$$\beta_{mn}^{(l)} = L_n + \sum_{m' \in \mathcal{M}(n)\backslash m} \alpha_{m'n}^{(l)} \tag{3.17}$$

3.3 グループ分割手法の関連研究

GSBP 復号法のグループ分けでは，受信語のビットを受信した順に，対応する変数ノードの添字をグループに振り分けている．しかし，SBP 復号法，GSBP 復号法では，先に更新するメッセージの信頼性を向上させることで，後に更新するメッセージの信頼性能が向上する．そのため，変数

ノードの添字を適切なグループに振り分けることにより，更新するメッセージの信頼性を向上させ，復号性能の改善を図ることが可能である．本節では，グループ分割手法の関連研究について述べる．

3.3.1 次数が降順となるグループ分割手法

SBP復号法，GSBP復号法では，次数が大きい変数ノードは，次数の小さい変数ノードに比べて，一度の処理で交換できるメッセージが多いため，メッセージの信頼性も高くなる．そこで，SBP復号法において，変数ノードの次数が降順となる順に更新処理をする手法が内川らにより提案されており[11]，また，GSBP復号法においても内川らの手法が有効であることが佐藤らにより示されている[12].

内川らの手法を用いたグループ分けでは，グループに振り分けられていない変数ノードの添字の中で，最大次数を持つ変数ノードの添字を選択しグループに振り分ける．これにより，次数が降順となるよう変数ノードを処理し，従来のGSBP復号法に比べて，信頼性の高いメッセージ更新を行うことが可能である．

3.3.2 隣接関係を考慮したグループ分割手法

GSBP復号法では，反復l回目のメッセージ$\alpha_{mn}^{(l)}$を更新するを更新する際，出来るだけ反復l回のメッセージ$\beta_{mn'}^{(l)}$を利用することで信頼性の高いメッセージ更新を行っていた．いま，グループ分けのとき，同じチェックノードc_1に隣接している変数ノードv_1, v_2を同じグループに含まれるようにグループ分割する場合を考える．このとき，同じグループ内では並列にメッセージ更新を行うので，反復l回目のc_1からv_2へのメッセージ$\alpha_{12}^{(l)}$に，反復l回目のv_1からc_1へのメッセージ$\beta_{11}^{(l)}$を用いることが出来ない．そこで，佐藤らは，同じチェックノードに隣接する変数ノードが，できるだけ同じグループに含まれないようにグループ分割する手法を提案している[12].

いま，変数ノードの添字$n = 1, 2, \ldots, N$をr個のグループ$G_1, G_2, \ldots, G_r$に分割することを考える．ここで，タナーグラフの辺の総数をD_{max}，各グループに含まれる変数ノードの辺の合計をD_k $(k = 1, 2, \ldots, r)$とし，各変

数ノード v_n $(n = 1, 2, \ldots, N)$ の次数を $d(v_n)$ とする．佐藤らのアルゴリズムを次に示す．

ステップ 1 $G_1 = G_2 = \ldots = G_r = \{\}$（$\{\}$ は空集合）とする．$k := 1$ とする．

ステップ 2 次数が降順となる順に変数ノードの添字を一つランダムに選択し，次式を満たす限り G_k に振り分ける．

$$\sum_{i=1}^{k-1} |D_i| + |D_k| \leq k \frac{D_{max}}{r} \tag{3.18}$$

式 (3.18) を満たさないとき，$k := k + 1$ として，ステップ 2 を繰り返す．でなければ，$n := 1$ としてステップ 3 へ進む．

ステップ 3 $n \in G_k$ とし，次式を満たす変数ノードの添字 q をランダムに一つ選択する．

$$q = \underset{\substack{n' \in G_{k'} \\ k \neq k',\, d(v'_n) = d(v_n)}}{\arg\min} \left\{ \left| \bigcup_{i \in G_{k'}} \mathcal{M}(n) \cap \mathcal{M}(i) \right| + \left| \bigcup_{j \in G_k} \mathcal{M}(n') \cap \mathcal{M}(j) \right| \right\} \tag{3.19}$$

ステップ 4 ステップ 3 で選んだ q に対して，次式から η, η' を求める．

$$\eta = \left| \bigcup_{i \in G_k} \mathcal{M}(n) \cap \mathcal{M}(i) \right| + \left| \bigcup_{j \in G_{k'}} \mathcal{M}(q) \cap \mathcal{M}(j) \right| \tag{3.20}$$

$$\eta' = \left| \bigcup_{j \in G_{k'}} \mathcal{M}(n) \cap \mathcal{M}(j) \right| + \left| \bigcup_{i \in G_k} \mathcal{M}(q) \cap \mathcal{M}(i) \right| \tag{3.21}$$

もし，$\eta > \eta'$ ならば，グループの n と q を入れ替える．$\eta \leq \eta'$ ならば，入れ替えない

ステップ 5 もし，$n = N$ ならば終了．そうでなければ，$n := n + 1$ としてステップ 3 へ戻る．

第4章　局所的内径を考慮したグループ分割手法

4.1　提案法の考え

GSBP復号法では，グループ間で逐次的に処理を行うため，先に処理したグループで更新したメッセージを次のグループ以降で利用し，収束速度を向上させていた．しかし，先に処理するグループで誤りを含むメッセージ更新を行った場合，従来のBP復号法に比べて，誤りを含むメッセージを早く伝播させてしまうため，解への収束が遅れてしまう問題がある．GSBP復号法[9]では，受信したビットの添字を昇順にグループに振り分けており，誤りの伝播については考慮していない．そのため，誤りを含むメッセージの伝播を防ぐことで，収束速度の向上が期待できる．

変数ノードが送るメッセージの信頼度の指標として，2章で述べた局所的内径がある．タナーグラフが局所的内径の小さい変数ノードを持つ場合，BP復号法の性能が悪化することが知られている[13]．実際，ある変数ノードを始点と終点とするような閉路がある場合，変数ノードからのメッセージが閉路を通して変数ノード自身に返ってくるため事後確率が収束しない状況に陥る場合があり，特に局所的内径の値が小さいとき，その影響が顕著に表れる．

そこで，変数ノードの局所的内径が降順となるよう先に処理するようグループ分けすることで，誤りの伝播を防ぐことができるのではないかと考えられる．また非正則LDPC符号の場合には，変数ノードの次数が降順となるようグループ分けをすることでことで収束速度が向上することが知られている[3]．本研究では，次数と局所的内径を考慮したグループ分割手法を提案する．

4.2 提案法のアルゴリズム

今，符号長Nの符号語のビットの添字を，サイズが均等である（つまり，各グループのサイズが$N_G = N/r$となる）r個のグループ$G_1, G_2, \ldots, G_r$に分割することを考える．また，各変数ノードの次数を$d(v_n)$ $(n = 1, 2, \ldots, N)$，局所的内径を$g(v_n)$ $(n = 1, 2, \ldots, N)$とする．ただし，どの閉路にも含まれない変数ノードの局所的内径は任意に大きな値とする．以下に提案法のアルゴリズムを示す．

提案法のアルゴリズム

ステップ１ $G_1 = G_2 = \ldots = G_r = \{\}$とする．$k := 1$とする

ステップ２ $|G_k| = N_G$となるまで，次の操作を繰り返す．

$n = 1, 2, \ldots, N$について，次の式(4.1)を求める．

$$\mathcal{A} = \left\{ n \notin \bigcup_{i=1}^{k} G_i \,\middle|\, d(v_n) = \max_{j \notin \bigcup_{j=1}^{k} G_j} d(v_j) \right\} \tag{4.1}$$

求めた$\mathcal{A}$について，次式(4.2)を求める．

$$\mathcal{B} = \{ n \in \mathcal{A} \mid g(v_n) = \max_{h \in \mathcal{A}} g(v_h) \} \tag{4.2}$$

$n^* = \min \mathcal{B}$をG_kに追加する．

ステップ３ $k = r$ならば終了．そうでなければ，$k := k + 1$として，ステップ２へ戻る．

提案法のステップ１は，各グループ$G_1, G_2, \ldots, G_r$を空集合とし，G_1から添字を振り分けていくことを意味する．ステップ２では，グループG_kの要素数が最大$|N_G|$になるまで，次の1)と2)の操作を繰り返す．

1) まだグループに振り分けられていない変数ノードの添字の中から，次数が最大となる添字を全て列挙し，それらの集まりを集合$\mathcal{A}$とおく（式(4.1)）．

2) $\mathcal{A}$に含まれる要素を添字として持つ変数ノードを考える．そのなかで，最大の局所的内径を持つ変数ノードの添字集合を$\mathcal{B}$とする（式(4.2)）．$\mathcal{B}$の中で最も小さい添字n^*をG_kに追加する．

ステップ 3 は終了条件で，全ての変数ノードのグループ分けが終了しているならばアルゴリズムを終了し，まだグループ分けされていない変数ノードが残っているならば，次のグループ G_{k+1} に添字を振り分けていく．

提案法を用いることで，次数は降順に，同じ次数同士であれば局所的内径が大きい変数ノードの添字から先に処理できる．これにより，信頼度が低く，誤る可能性が高い変数ノードのメッセージの伝播を防ぐことができる．

4.2.1 提案法の例

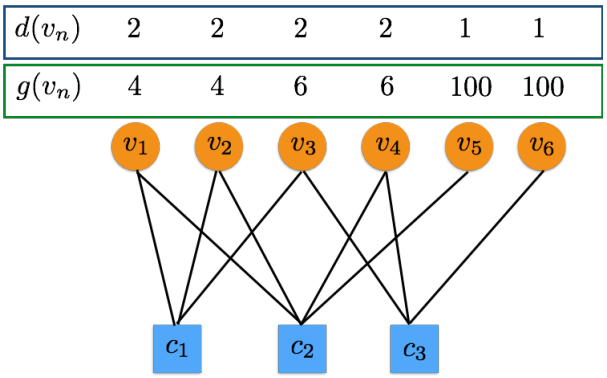

図 4.1: 提案法のグループ分けの例のタナーグラフ

図 4.1 のタナーグラフを用いて提案法の例を示す．今，変数ノード $v_1, \ldots, v_6$ の添字を 3 つのグループ G_1, G_2, G_3 に分けるとする（よって，各グループに入る要素数は $N_G = 6/3 = 2$ となる）．どの閉路にも含まれない変数ノードの局所的内径の値は 100（任意の大きな値）とする．図 4.1 の例では，各変数ノードの次数は $d(v_1) = d(v_2) = d(v_3) = d(v_4) = 2$, $d(v_5) = d(v_6) = 1$, また，局所的内径は $g(v_1) = 4, g(v_2) = 4, g(v_3) = 6, g(v_4) = 6, g(v_5) = 100, g(v_6) = 100$ である．

提案法の例を次に示す．最初に，ステップ 1 で各グループを $G_1 = G_2 = G_3 = \{\}$ とする．次に，ステップ 2 で $|G_1| = 2$ となるまで，変数ノードの添字を振り分ける．まず，式 (4.1) により，最大次数の変数ノード集合 $\mathcal{A}$ を求める．次数が最大となるのは，次数が 2 である v_1, v_2, v_3, v_4 となるので，添字集合 $\mathcal{A} = \{1, 2, 3, 4\}$ を得る．次に，$\mathcal{A}$ の中で，局所的内径が最大の変数

ノードの添字集合$\mathcal{B}$を求めると，$\mathcal{B}=\{3,4\}$となる．そして，$\mathcal{B}$の要素の中で添字の値が最小になる3がG_1に追加される（図4.2）．同様に，もう一度ステップ2を行う．まだグループに振り分けられていない変数ノードの中で最大次数となる変数ノードの添字集合$\mathcal{A}$を求めると，$\mathcal{A}=\{1,2,4\}$を得る．$\mathcal{A}=\{1,2,4\}$の中で最大の局所的内径を持つ変数ノードの添字は4なので，4をG_1に追加し，$G_1=\{3,4\}$を得る（図4.3）．$|G_1|=2$となるのでステップ3へ移る．ステップ3では，グループ分けが終了していないので，$k:=k+1$とし，G_2へ添字を振り分けるためにステップ2へ戻る．今度はG_2に対してステップ2を行い，$G_2=\{1,2\}$を得る．ステップ3で$k:=k+1$としステップ2へ戻る．最後に，ステップ2でG_3に対して振り分けを行い$G_3=\{5,6\}$となる．これで，すべての変数ノードの添字がグループに振り分けられた（図4.4）．

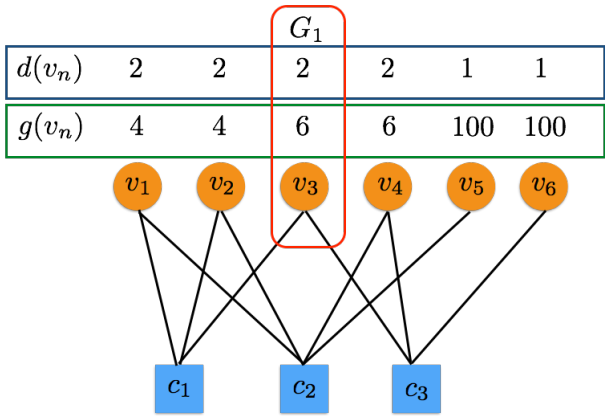

図 4.2: 提案法のグループ分けの例 1

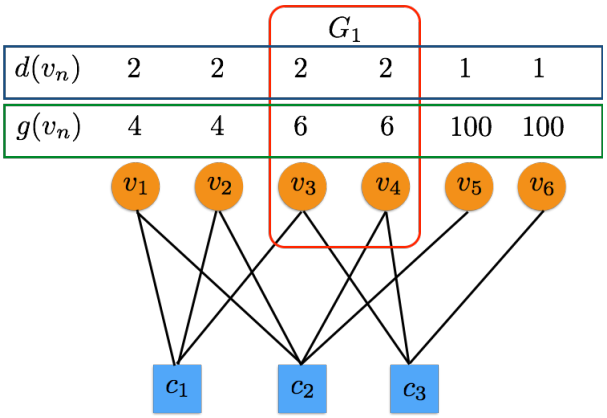

図 4.3: 提案法のグループ分けの例 2

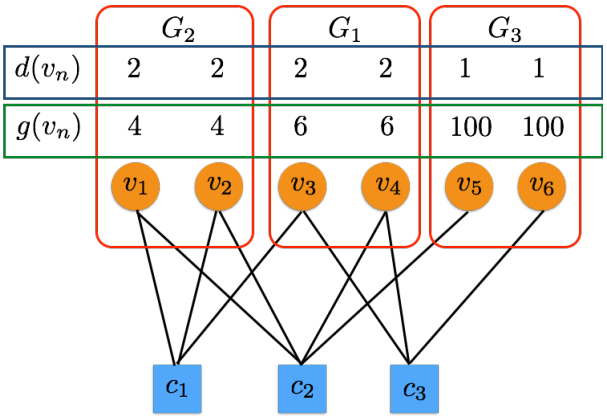

図 4.4: 提案法のグループ分けの例 3

第5章　計算機実験と考察

5.1　実験環境

提案法の効果を評価するために，計算機実験によるシミュレーションを行った．

用いた符号は，符号長$N = 1024$，符号化率$R = 1/2$のIRA符号で，文献[5]より変数ノードの次数分布を$\lambda(x) = 0.333x + 0.3851x^2 + 0.0002x^3 + 0.1392x^6 + 0.1425x^7$，チェックノードの次数分布を$\rho(x) = 0.9849x^5 + 0.0151x^6$とした．また，検査行列$\boldsymbol{H}$の部分行列$\boldsymbol{H}_u$は3.1.2節で述べた立川の構成法を用いて構成した．符号の次数と局所的内径の組み合せと対応する変数ノードの個数を表5.1に示す．

復号法のプログラム実装に関しては，和田山が公開しているBP復号法のプログラム[14]を参考にしてGSBP復号法に改良した．

実験では，既存法と提案法の性能評価を行った．既存法として，3.3.2節で述べた，次数が降順となる順に変数ノードの添字をグループに振り分ける手法を用いた．

評価した内容は，横軸をSNRとしたときのビット誤り率，改善率，平均反復回数である．ここで，改善率は，

$$\text{改善率} = \frac{\text{既存法のビット誤り率} - \text{提案法のビット誤り率}}{\text{既存法のビット誤り率}} \times 100$$

と定義し，平均反復回数は，推定語が符号語となるまで復号するのにかかった反復回数の平均値である．実験では，AWGN通信路を仮定し，推定語でのビットの誤りが10^5回発生する，もしくは試行回数が10^7回になるまで試行する．

また，横軸を最大反復回数としたときの既存法と提案法のビット誤り率の変化も評価した．実験では，AWGN通信路を仮定し，試行回数を10^6回とする．

表 5.1: 用いた符号の変数ノードの次数と局所的内径

次数	局所的内径	ノード数
1	100	1
2	4	11
2	6	166
2	8	329
2	10	5
3	4	21
3	6	204
3	8	170
4	4	1
7	4	14
7	6	47
8	4	25
8	6	30

5.2 実験結果と考察

5.2.1 ビット誤り率と平均反復回数

最大反復回数5回のビット誤り率

最大反復回数5回，グループ数$r = 4, 8, 16$の既存法と提案法のビット誤り率を図5.1に示す．図の横軸はSNR，縦軸はビット誤り率を表す．また，図中の黒い破線GSBP($r = 1$)は通常のBP復号法を表し，また，破線(GSBP)が既存法，実線(Proposed)が提案法で，同じ色が同じグループ数rを表す．図5.2は図5.1のSNR 4.0[dB]から5.0[dB]を拡大した図である．

図5.1，5.2より，既存法，提案法ともにグループ数rが$4, 8, 16$と増えるに従いビット誤り率が改善していることが分かる．これは，グループ数が増加すると，より多くの更新メッージが利用できるためである．また，各グループ数で既存法と提案法を比較すると，提案法のビット誤り率が既存法よりも改善されている．例として，SNR $= 4.5$[dB]での，グループ数$r = 16$の提案法と既存法を比較すると，約4.5×10^{-7}（改善率で36%）の改善が見られる．同じグループ数での提案法による改善は，提案法を用いることで誤ったメッセージの伝播を防ぎ，信頼性の高いメッセージ更新が行えているためと考えられる．

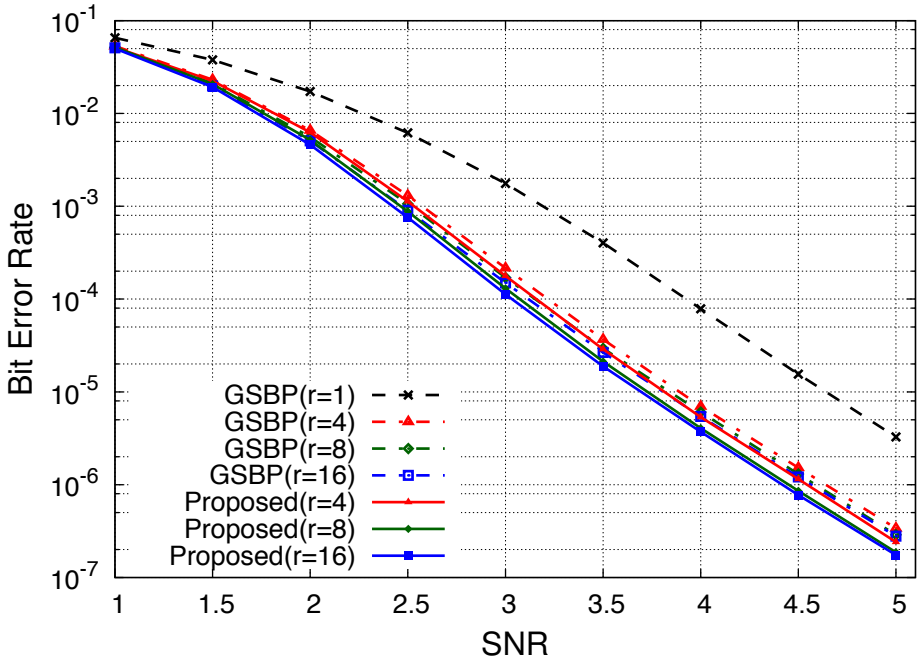

図 5.1: ビット誤り率（最大反復回数5回，グループ数$r = 4, 8, 16$）

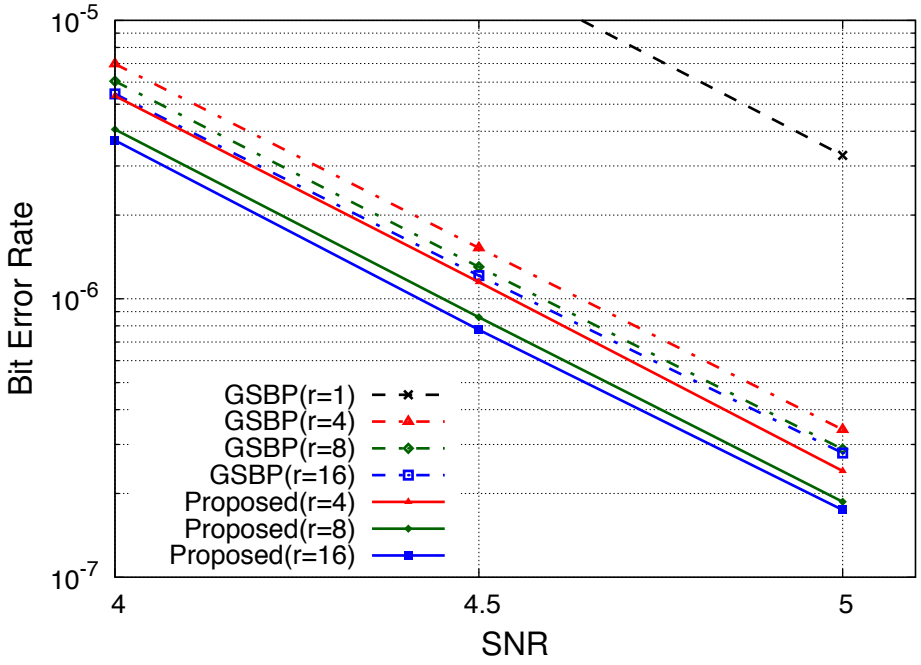

図 5.2: ビット誤り率（最大反復回数5回，グループ数$r = 4, 8, 16$）の拡大図

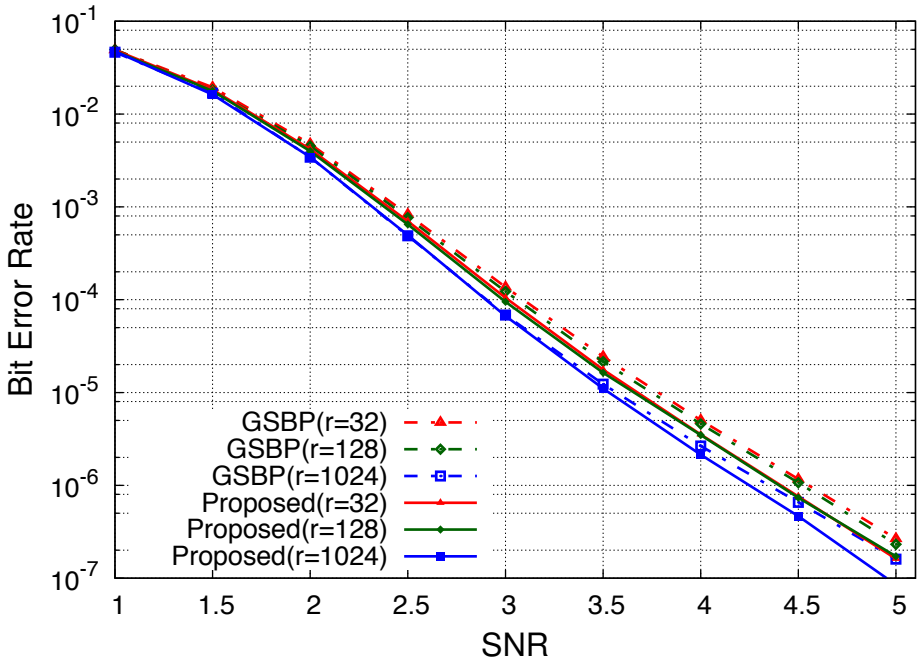

図 5.3: ビット誤り率（最大反復回数5回，グループ数$r = 32, 128, 1024$）

さらに，最大反復回数5回，グループ数$r = 32, 128, 1024$の既存法と提案法のビット誤り率を図5.3に示す．破線(GSBP)が既存法，実線(Proposed)が提案法で，同じ色が同じグループ数rを表す．特に，グループ数$r = 1024$はSBP復号法となる．図5.4は図5.3のSNR 4[dB]から5[dB]を拡大した図である．

図5.3，5.4より，グループ数$r = 4, 8, 16$と同様に，既存法，提案法ともにグループ数rが$32, 128, 1024$と増えるに従いビット誤り率が改善していることや，各グループ数で既存法と提案法を比較した際の改善も見られる．特に，SNR= 5[dB]のときのビット誤り率では，提案法のグループ数$r = 32, 128$が既存法で最もビット誤り率が低いSBP復号法（GSBP($r = 1024$)）と同等のビット誤り率を示しており，最大反復5回では，提案法により少ないグループ数で従来のSBP復号法の訂正能力が得られた．

最大反復回数5回の改善率

図5.5にグループ数$r = 4, 8, 16$の，図5.6にグループ数$r = 32, 128, 1024$の提案法を用いることによる改善率を示す．横軸はSNR，縦軸は改善率を表す．図5.5中の$r = 4$，$r = 8$，$r = 16$がそれぞれグループ数$r = 4, 8, 16$に，図5.6中の$r = 32$，$r = 128$，$r = 1024$がそれぞれグループ数$r = 16, 128, 1024$に

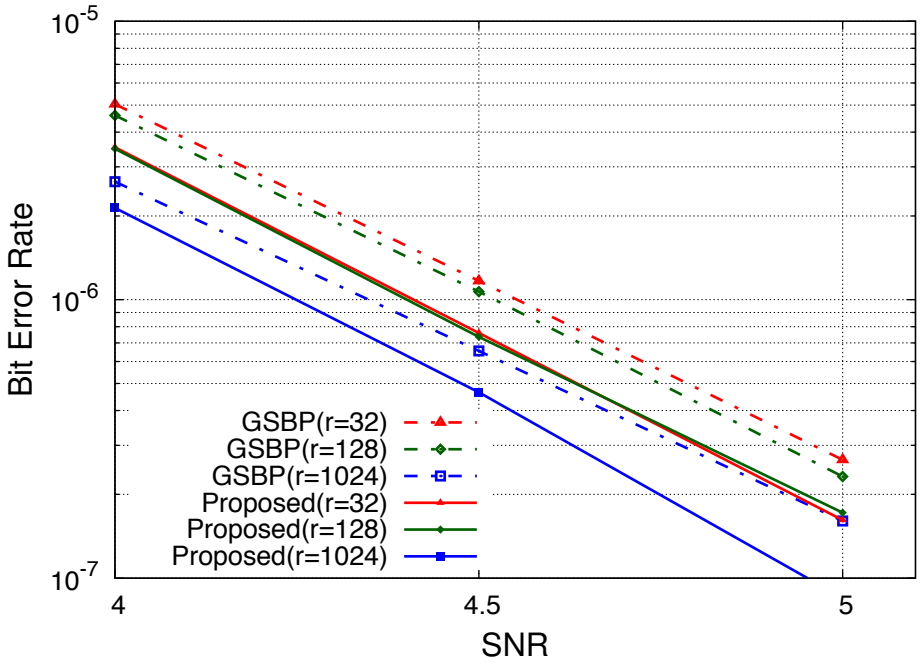

図 5.4: ビット誤り率（最大反復回数5回，グループ数$r = 32, 128, 1024$）の拡大図

対応している．

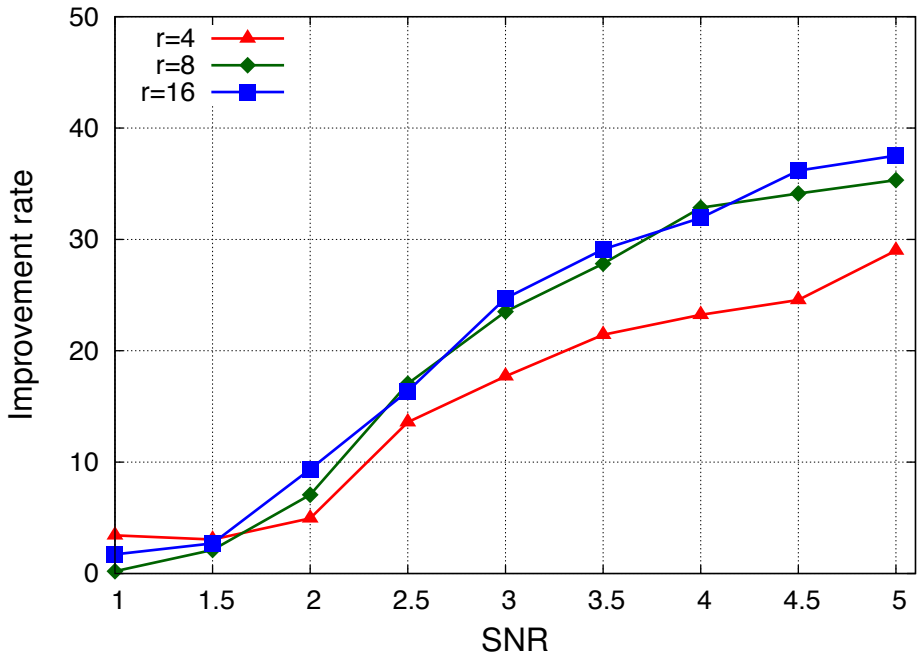

図 5.5: 提案法の改善率（最大反復回数5回，グループ数$r = 4, 8, 16$）

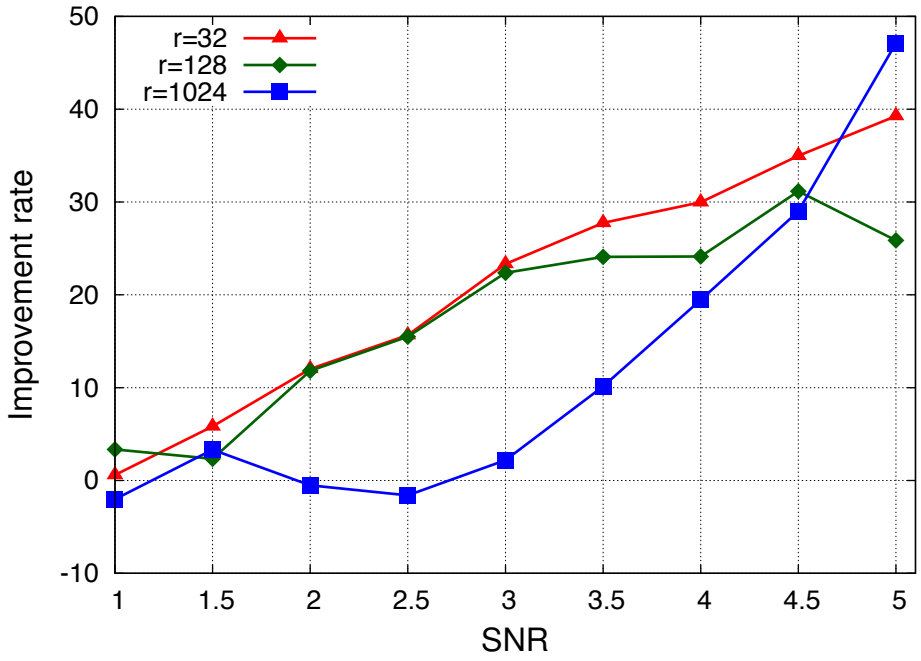

図 5.6: 提案法の改善率（最大反復回数5回，グループ数$r = 32, 128, 1024$）

図5.5より，グループ数$r = 4, 8, 16$では，全てのSNRで提案法により改善したことがわかる．例えば，SNR= 4.5[dB]でのグループ数$r = 16$の場合，36%の改善が見られる．SNRが変化したときの改善率を見ると，SNRが増加するに従い各グループの改善率も上昇するが，一方で，SNR= 3.0[dB]以降は改善率の変化の傾きが緩やかになる傾向が見られる．また，グループ数ごとの改善率を比較すると，グループ数$r = 4$の改善率に比べグループ数$r = 8, 16$の改善率は高くなるが，グループ数$r = 8, 16$では，改善率に大きな差は見られないことが分かる．

図5.6より，グループ数$r = 32, 128$でも全てのSNRで提案法により改善したことがわかる．また，SNRが変化したときの改善率の変化より，グループ数$r = 32, 128$では，SNRが1.0[dB]から4.0[dB]にかけてグループ数$r = 4, 8, 16$と同様に改善率の傾きが緩やかになる傾向が見られることがわかった．一方で，グループ数$r = 1024$の改善率は，SNR 3.0[dB]以降で急激に上昇しており，その原因の調査は今後の課題とする．

最大反復回数5回の平均反復回数

図5.7にグループ数$r = 4, 8, 16$の，図5.8にグループ数$r = 32, 128, 1024$の既存法と提案法の平均反復回数を示す．

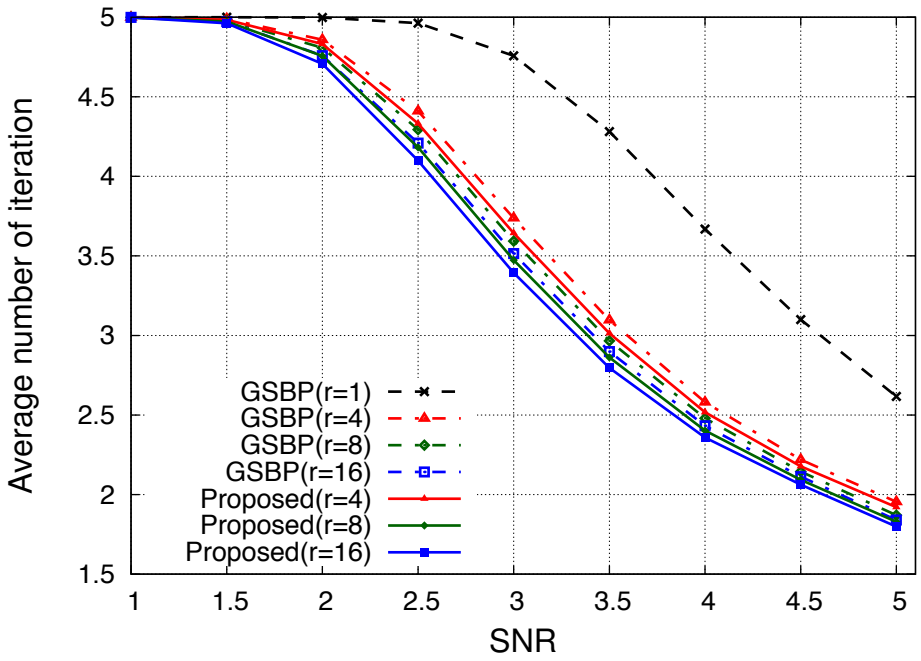

図 5.7: 平均反復回数（最大反復回数5回，グループ数$r = 4, 8, 16$）

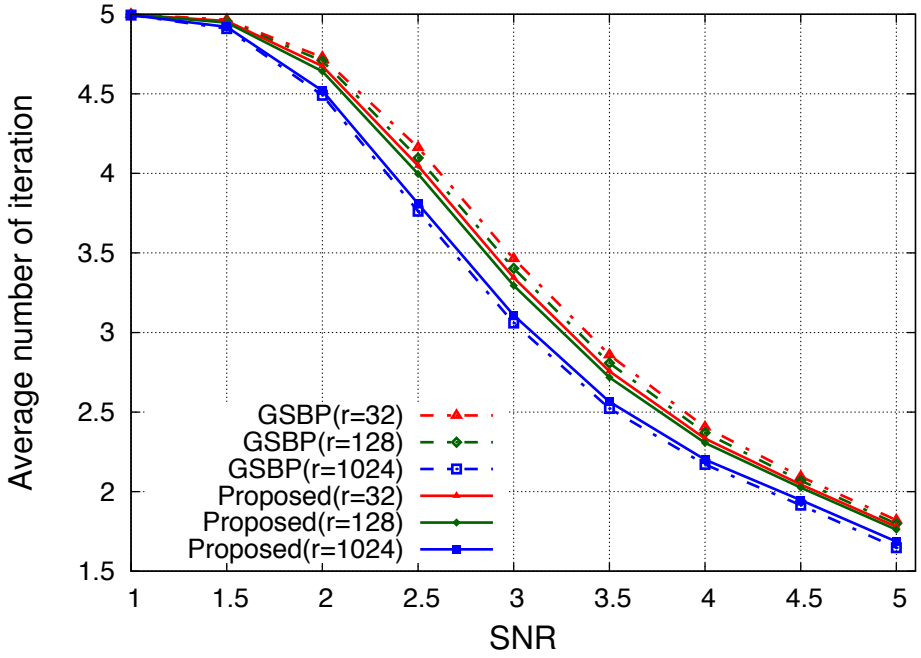

図 5.8: 平均反復回数（最大反復回数5回，グループ数$r = 32, 128, 1024$）

図5.7，図5.8より，グループ数$r = 4, 8, 16, 32, 128$において既存法に比べ提案法の平均反復回数が減少していることがわかる．最大反復回数5回でのビット誤り率が既存法に比べ提案法が優れていたことから，最大反復回数5回のグループ数$r = 4, 8, 16, 32, 128$では，提案法により少ない反復回数で復号が成功しており，収束速度が向上していると言える．一方で，グループ数$r = 1024$の場合は，提案法が既存法とほぼ同等の平均反復回数となった．

最大反復回数10回のビット誤り率

最大反復回数10回，グループ数$r = 4, 8, 16$の既存法と提案法のビット誤り率を図5.9に示す．図5.10は図5.9のSNR 1.5[dB]から2.5[dB]を拡大した図，図5.11は図5.9のSNR 3.5[dB]から4.5[dB]を拡大した図である．

図5.10より，SNRが2.0[dB]前後では提案法によりグループ数$r = 4, 8, 16$のビット誤り率の若干の改善が見られた．これは，提案法により信頼性の高いメッセージ更新が行われたためと考えられる．しかし，図5.11より，SNR 3.5[dB]から4.5[dB]までては，既存法に比べ提案法のビット誤り率が高くなっており，性能が劣化していることがわかる．例えば，SNR 4.0のグループ数$r = 8$のビット誤り率では，既存法に比べ提案法の方が6.0×10^{-7}

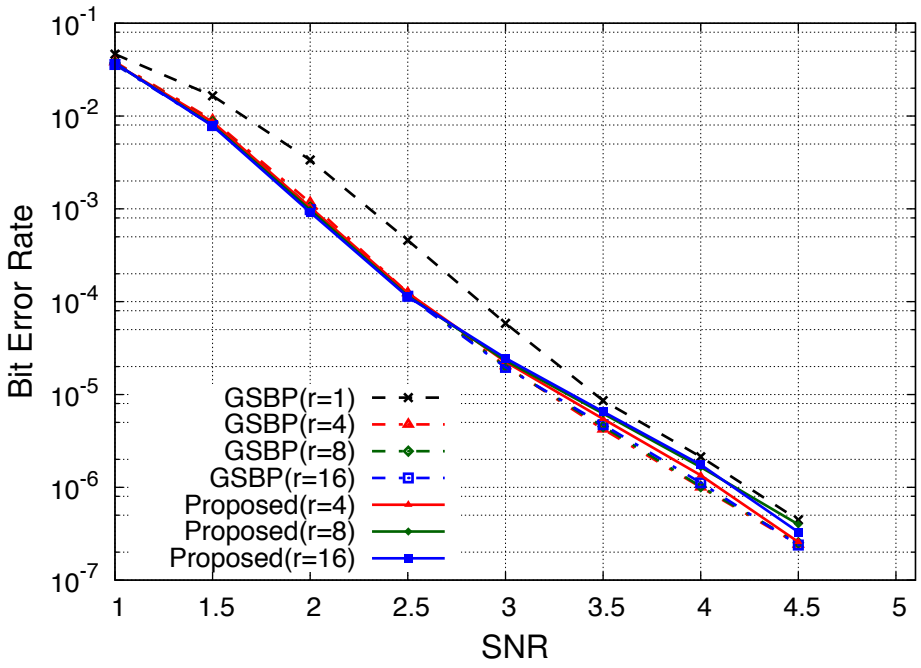

図 5.9: ビット誤り率（最大反復回数 10 回，グループ数 $r = 4, 8, 16$）

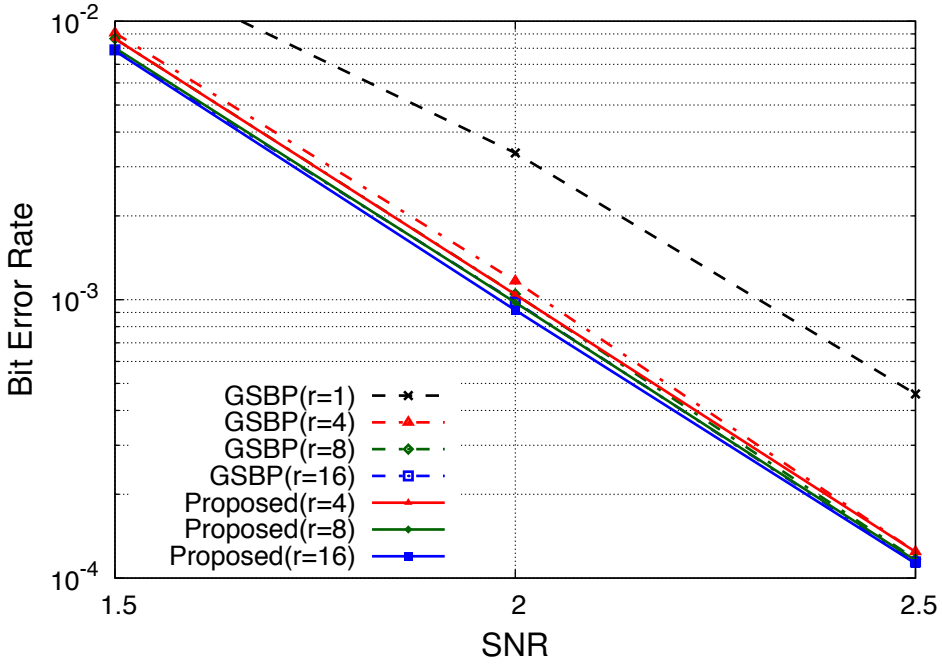

図 5.10: ビット誤り率（最大反復回数 10 回，グループ数 $r = 4, 8, 16$）の SNR 1.5[dB] から 2.5[dB] までの拡大図

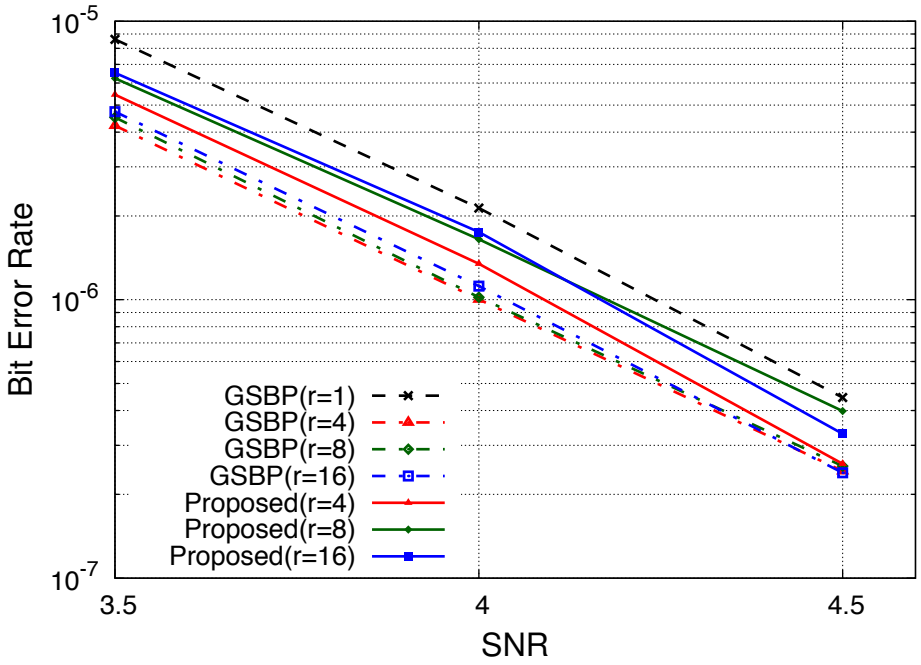

図 5.11: ビット誤り率（最大反復回数 10 回，グループ数 $r = 4, 8, 16$）の SNR 3.5[dB] から 4.5[dB] までの拡大図

増加している．この原因としては，提案法により同じチェックノードと隣接する変数ノードを同じグループに振り分けてしまい，更新メッセージを利用する回数が既存法に比べ減ったためと考えられる．提案法では，同じ次数のときには局所的内径が最大となる変数ノードからグループに振り分けるが，同じ閉路に含まれる変数ノードの局所的内径は同じ値になる可能性が高く，同じグループに振り分けれてしまう可能性も高くなる．このとき同じグループに属する変数ノードが同じチェックノードと隣接する確率も高くなるが，3.3.2 節で述べた通り，同じチェックノードと隣接する変数ノードが同じグループに振り分けられると更新したメッセージが利用できない問題点があるため，性能が劣化したと考えられる．

最大反復回数 10 回では，SNR 2.5[dB] までは提案法によりビット誤り率が改善し，3.0[dB] 以降は劣化した．これまで議論より，反復回数 10 回程では，SNR が小さく雑音が多い場合に，隣接関係の影響に比べ局所的内径を考慮した誤りの伝播を防ぐ手法の効果が高く，SNR が大きくなるに従い，隣接関係による更新回数の変化の影響が強くなることが考えられる．

最大反復回数 10 回，グループ数 $r = 32, 128, 1024$ の既存法と提案法のビット誤り率を図 5.12 に示す．また，図 5.13 は図 5.12 の SNR 3.5[dB] から 4.5[dB]

を拡大した図である．

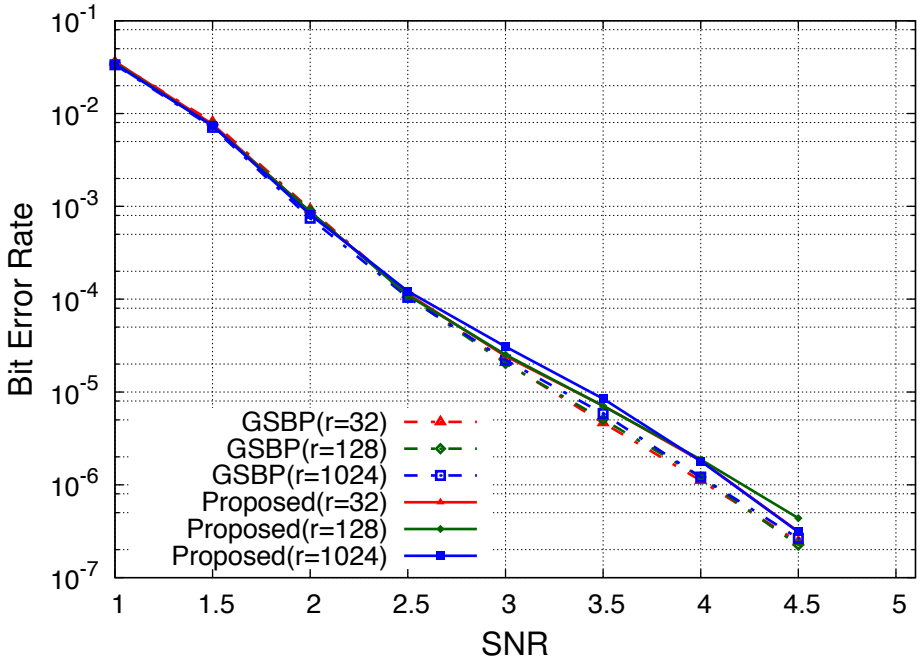

図 5.12: ビット誤り率（最大反復回数 10 回，グループ数 $r = 32, 128, 1024$）

図 5.13 より，SNR が大きい場合，既存法に比べ提案法のビット誤り率が高くなっている．グループ数 $r = 32, 128$ の場合の誤り訂正能力の劣化は，図 5.10 と同様に，提案法により同じチェックノードと隣接した変数ノードが同じグループに振り分けられてしまうことが原因と考えられる．一方で，グループ数 $r = 1024$ の場合，すべての変数ノードで逐次実行しているため，図 5.10 と同様の議論は成り立たない．そのため，SNR が大きい場合に局所的内径が降順になるようにグループに振り分けた場合，誤り訂正能力劣化に影響を与える要因が隣接関係とは別に存在する可能性がある．

最大反復回数 10 回の改善率

図 5.14 にグループ数 $r = 4, 8, 16$ の，図 5.15 にグループ数 $r = 32, 128, 1024$ の提案法を用いることによる改善率を示す．

図 5.14，図 5.15 より，グループ数 $r = 1024$ の場合を除き，全てのグループ数で SNR が 2.5[dB] まで改善が見られる．例えば，SNR 2[dB] でのグループ数 $r = 8$ の場合，提案法により 10% の改善が見られる．また，2.5[dB] から SNR 4.5[dB] まで改善率が負となり，性能が劣化することがわかる．こ

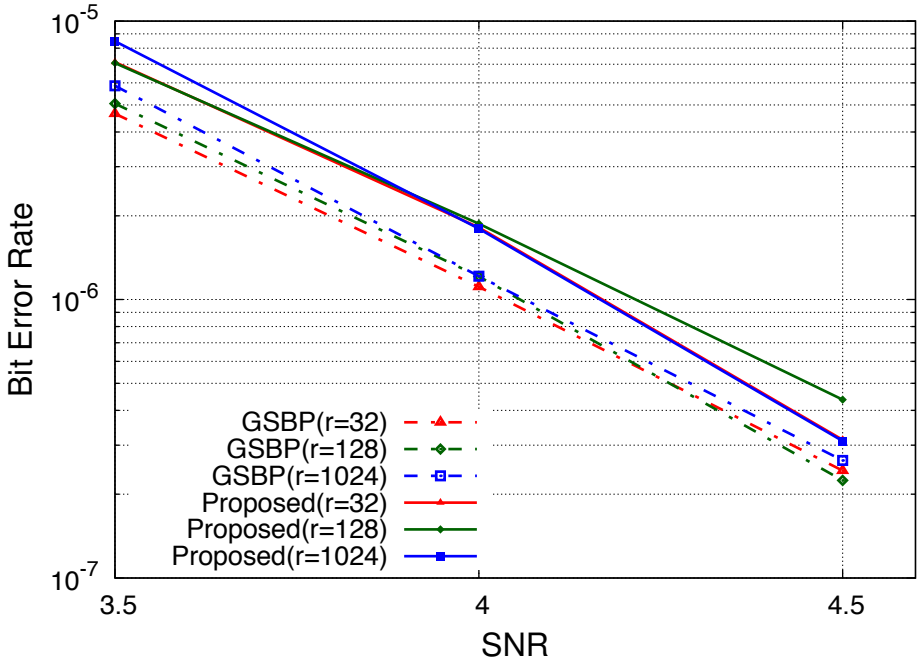

図 5.13: ビット誤り率（最大反復回数 10 回，グループ数 $r = 32, 128, 1024$）の拡大図

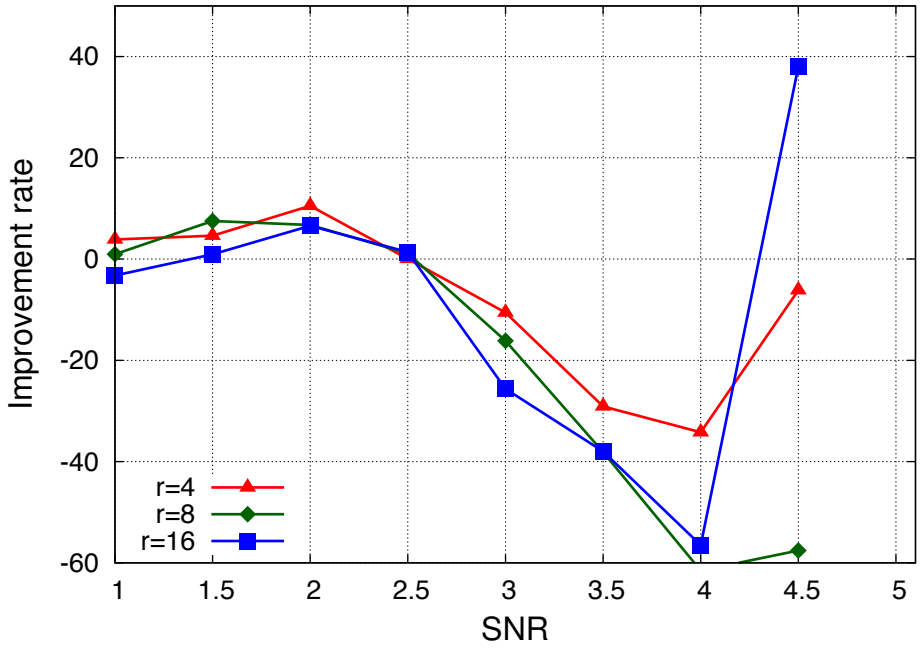

図 5.14: 提案法の改善率（最大反復回数 10 回，グループ数 $r = 4, 8, 16$）

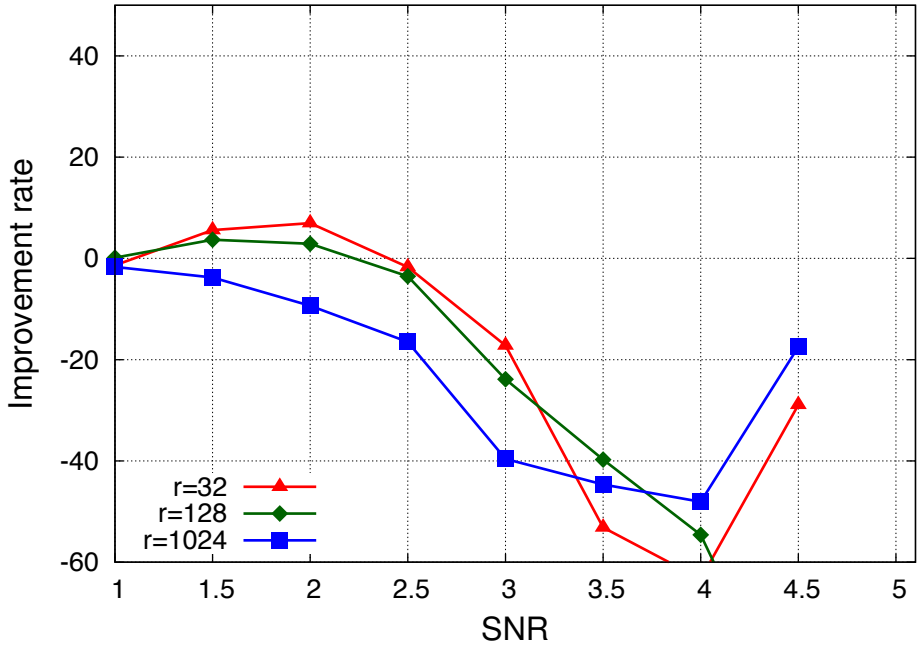

図 5.15: 提案法の改善率（最大反復回数 10 回，グループ数 $r = 32, 128, 1024$）

れは，先に述べた提案法と隣接関係の影響と考えられる．一方で，SNR 4.5[dB] では，改善率が急激に改善している．この原因の調査は今後の課題とする．

最大反復回数 10 回の平均反復回数

図 5.16 にグループ数 $r = 4, 8, 16$ の，図 5.17 にグループ数 $r = 32, 128, 1024$ の最大反復回数 10 回の既存法と提案法の平均反復回数を示す．

図 5.16，図 5.17 より，グループ数 $r = 1024$ の場合を除き，全てのグループで既存法に比べ提案法の平均反復回数が減少している．SNR $= 2.5$[dB] までは，提案法により誤りの伝播を防ぐことで，早く解へ収束しているため平均反復回数が減少している．一方で，SNR$= 3.0$[dB] 以降も平均反復回数が減少している．最大反復回数 10 回のビット誤り率の結果より，訂正能力が劣化しているにもかかわらず，平均反復回数が減少していることより，送信した符号語とは異なる符号語に収束して，復号を終了していることが考えられる．

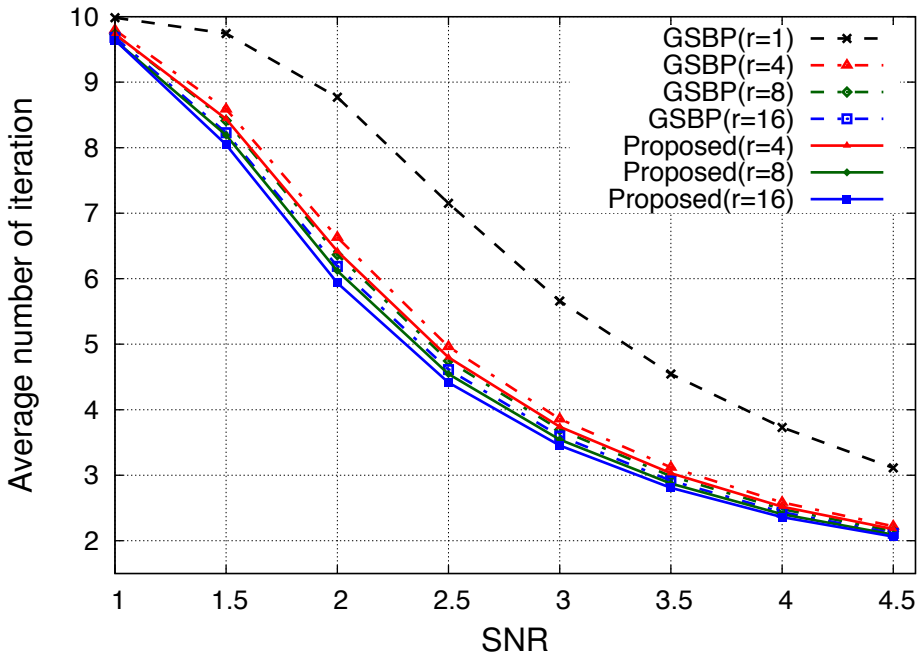

図 5.16: 平均反復回数（最大反復回数 10 回，グループ数 $r = 4, 8, 16$）

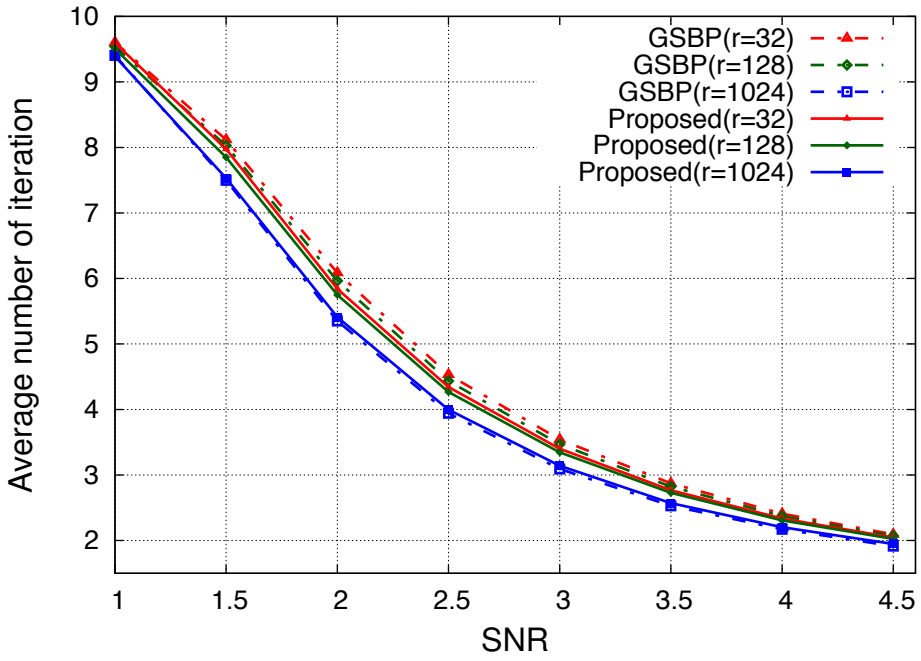

図 5.17: 平均反復回数（最大反復回数 10 回，グループ数 $r = 32, 128, 1024$）

最大反復回数20回のビット誤り率

最大反復回数20回，グループ数$r = 4, 8, 16$の既存法と提案法のビット誤り率を図5.18に，グループ数$r = 32, 128, 1024$の既存法と提案法のビット誤り率を図5.19に示す.

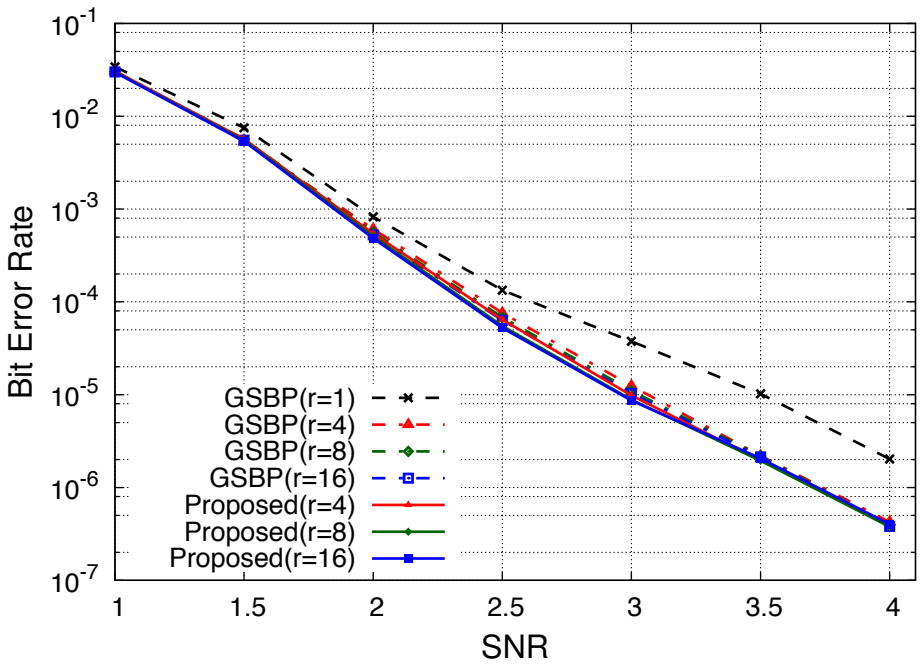

図 5.18: ビット誤り率（最大反復回数20回，グループ数$r = 4, 8, 16$）

図5.18，図5.19より，BP復号法を除いた場合，既存法と提案法がともに，どのクループ数でも同等の性能を示す．これより，十分な反復回数では，既存法も提案法も同等の訂正能力を持つことが考えられる.

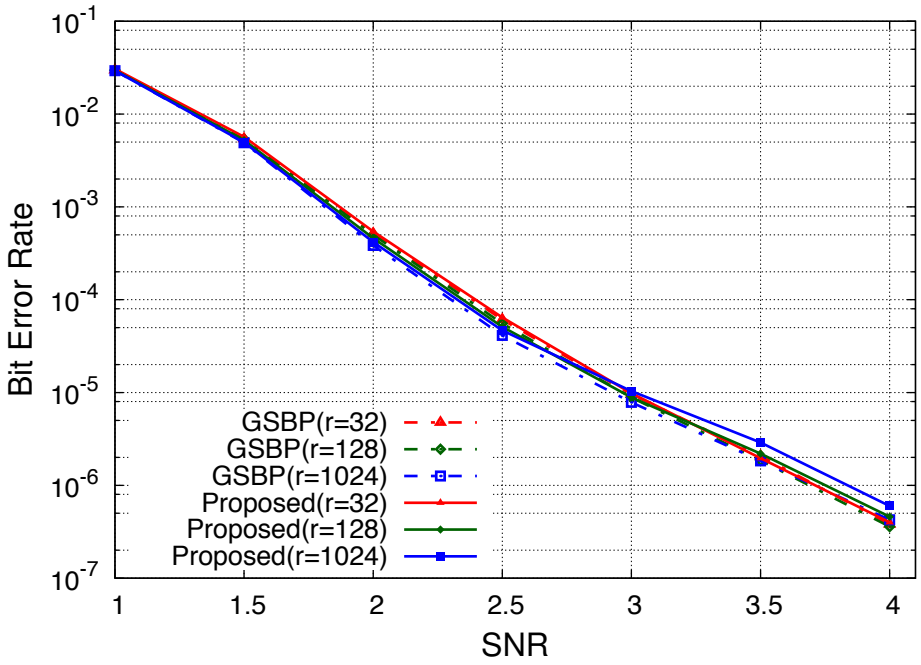

図 5.19: ビット誤り率（最大反復回数 20 回，グループ数 $r = 32, 128, 1024$）

5.2.2 最大反復回数の変化に伴うビット誤り率の変化

SNR が 2.5[dB] 以降では，最大反復回数 5 回では，既存法に比べ提案法のビット誤り率が低くなり，一方で，最大反復回数 10 回では，既存法のほうがビット誤り率が低くなった．そこで，SNR を 3[dB] に固定した場合，最大反復回数を変化させたときにビット誤り率がどのように変化するかをグループ数 $r = 4, 8, 16$ で調べた．

最大反復回数を 1 から 20 まで変化させたときのグループ数 $r = 4, 8, 16$ のビット誤り率をそれぞれ図 5.20，5.21，5.22 に示す．

図 5.20，5.21，5.22 より，どのグループ数でも，既存法も提案法も最大反復回数 8 回付近までは，急激にビット誤り率が低くなる．その後，8 回以降で一旦ビット誤り率が緩やかに高くなり，そして，再度減少していくことがわかる．また，どのグループでも，最大反復回数が 8 回までは，既存法に比べ提案法のビット誤り率が低くなることがわかる．

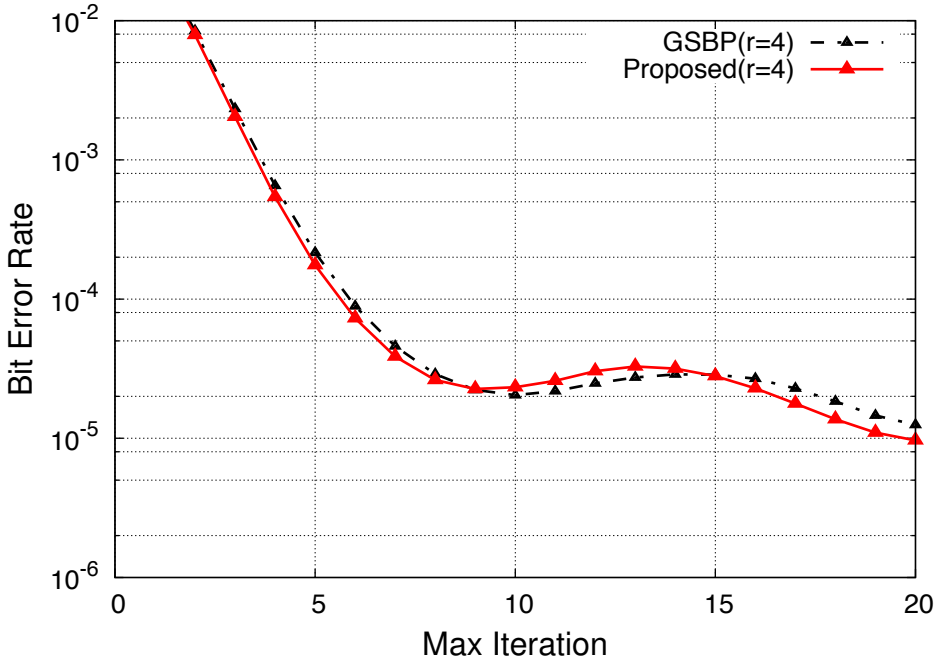

図 5.20: 既存法と提案法の最大反復回数ごとのビット誤り率（SNR=3，グループ数$r = 4$）

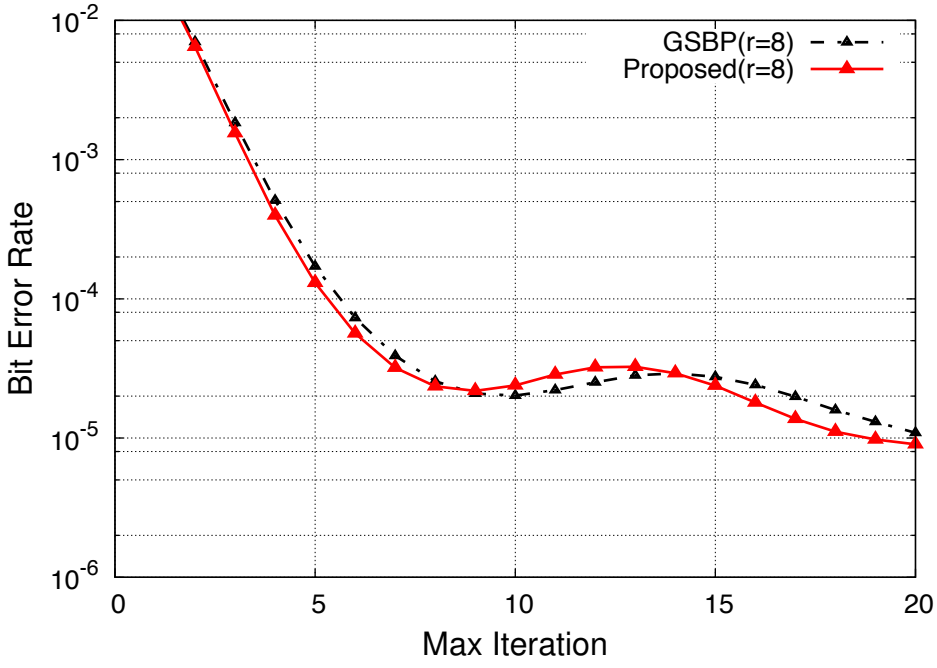

図 5.21: 既存法と提案法の最大反復回数ごとのビット誤り率（SNR=3，グループ数$r = 8$）

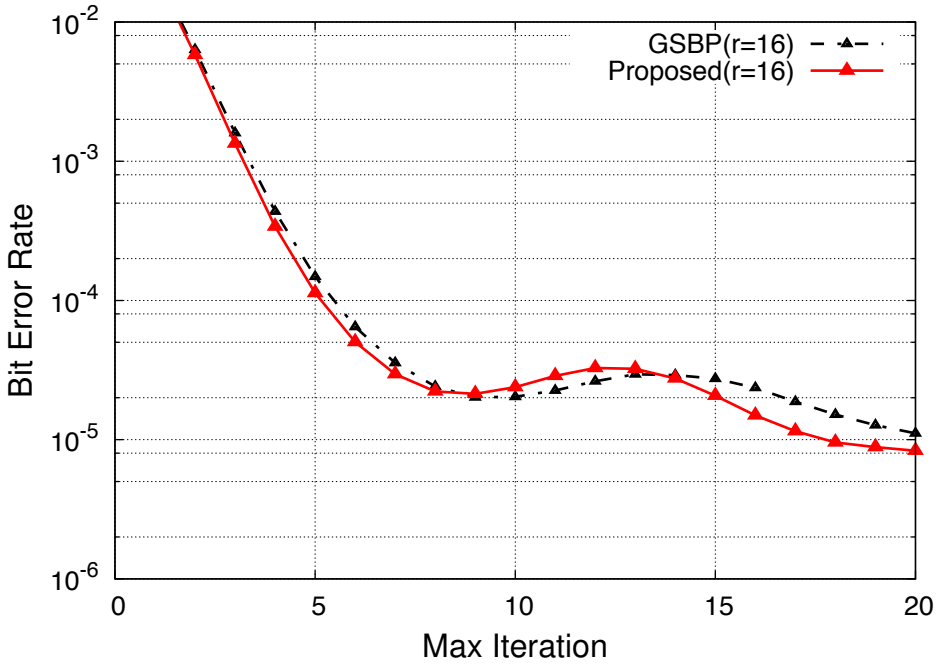

図 5.22: 既存法と提案法の最大反復回数ごとのビット誤り率（SNR=3，グループ数$r = 16$）

第6章　まとめと今後の課題

6.1　まとめ

本論文では，GSBP復号法の新しいグループ分割手法として，局所的内径を指標としたグループ分割手法を提案した．

GSBP復号法は，変数ノードを複数のグループに分割し，グループ内では並列処理，グループ間では逐次処理することで，遅延を減らしつつ収束速度を上げる手法である．GSBP復号法では，グループの分割手法により異なる性能を示すことが知られている．

そこで本論文では，局所的内径に注目した．局所的内径の値が小さい変数ノードにおいて，復号のメッセージ更新が正確に出来ず解へ収束しない場合がある．そこで，提案法では，局所的内径が降順になるようグループに振り分けることで誤りの伝播を防ぐことを考えた．また，既存研究である次数が降順になるようにグループに振り分ける手法の考えも取り入れ，次数が降順，かつ同じ次数内で局所的内径が降順となるようグループ分けする手法を提案した．

5章では，計算機実験により提案法の評価を行った．実験結果より，最大反復回数5回では，提案法のほうが従来法に比べ少ないグループで良いビット誤り率となることを示した．例として，SNR 4.5[dB]では，グループ数が16の提案法と既存法では，提案法には約4.5×10^{-7}の改善が見られ，改善率でも，36%の改善を示した．また，各グループの既存法と提案法の平均反復回数を比較し，従来法と比べて提案法のほうが収束速度が向上したことも示した．一方で，最大反復回数10回の実験では，SNR 2.5[dB]まででは提案法による改善がしたが，SNR 3.0[dB]以降では，既存法に比べ提案法の性能が劣化が見られた．この原因として，局所的内径の性質上，同じ局所的内径を持つ変数ノードが同じグループに含まれる可能性が高くなり，その結果，更新メッセージの利用回数が既存法に比べ減少したためではないかと考えられる．

6.2 今後の課題

本論文では，GSBP復号法のグループ分割にこれまで用いられなかった局所的内径を取り入れ，復号性能への影響を評価した．今後の課題としては，まず，反復回数が10回の場合の性能の劣化の原因を調査する．また，変数ノードの次数や局所的内径を独立に考えるのではなく，ノード間の隣接関係を含めて性能を評価し，局所的内径と隣接関係が性能に与える影響について調査すること，およびその調査結果を生かした新しいグループ分割手法を提案することが挙げられる．

謝辞

本論文の執筆に際し，ご指導，ご教示賜った森田 啓義教授，小川 朋宏准教授，山口 和彦准教授，笠井 裕之准教授，眞田 亜紀子助教に心より深く感謝いたします. 特に，眞田 亜紀子助教には，多くの時間をを指導や学会の準備等にあてて頂き，厚く御礼申しあげます.

本研究に際して，森田研究室の卒業生の立川 隆一氏には多くの有益な助言やプログラム等の技術的な支援を頂きました.また，日頃から研究にご協力頂いた応用ネットワーク講座の皆様方，特に，ゼミでの議論につきあって頂いた名和 拓朗氏，酒井 啓正氏，青木 健吾氏には感謝を申し上げます. 最後に，生活面で支援をして頂いた家族に感謝いたします.

参考文献

[1] R.G.Gallager, “Low-density parity-check codes,” in Research Monograph series. Cambridge, MIT Press, 1963.

[2] D. J. C. MacKay, “Good error-correcting codes based on very sparse matrices,” IEEE Trans. Inf. Theory, vol. 45, pp. 399–431, 1999.

[3] T. Richardson, M. Shokrollahi, and R. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” IEEE Trans. Inf. Theory, vol. 47, no. 2, pp. 619–637, 2001.

[4] H. Jin, A. Khandekar and R. McEliece, “Irregular repeat-accumulate codes,” Proc. Int. Symp. Turbo Codes Related Topics, pp. 1–8, 2000.

[5] S. Johnson and W. R. Steven, “Constructions for irregular repeat-accumulate codes,” Proc. IEEE Int. Symp. Inf Theory, pp. 179–183, 2005.

[6] D. B. West, “Introduction to Graph Theory (second edition),” Prentice Hall, 2000.

[7] Ryuichi TATSUKAWA, “Performance Evaluation of Short-length Irregular Repeat Accumulate Codes derived from Max-Flow Algorithm,” 修士学位論文, 電気通信大学, 2015.

[8] 和田山正「低密度パリティ検査符号とその復号法」株式会社トリケップス, 2002.

[9] J. Zhang and M. Fossorier, “Shuffled belief propagation decoding,” Proc. 36th Annu. Asilomar Conf. Signals, Syst., Computers, pp. 8–15, Nov. 2002.

[10] J. Zhang and F. Marc, “Shuffled iterative decoding,” IEEE Trans. Commun.,vol. 53, no. 2, pp. 209–213, 2005.

[11] 内川浩典, 原田康祐, 佐藤一美, “更新順序を決定するShuffled BP復号法,” 第29回情報理論とその応用シンポジウム予稿集, SITA 2006, vol. 29, no. 2, pp. 823-826, 2006.

[12] 佐藤芳行, 細谷剛, 平澤茂一, “Group Shuffled BP復号法に対する非正則LDPC符号の次数分布を考慮した効果的なグループ分割法,” 電子情報通信学会技術研究報告. IT, 情報理論 108(472), pp. 421–426, 2009

[13] Y. Mao and A. Banihashemi, “A heuristic search for good low-density parity-check codes at short block lengths,” IEEE Int. Conf., ICC, Commun., Vol. 1, pp. 41–44, 2001

[14] 和田山正, 「LDPC符号関係の公開プログラム」, https://dl.dropboxusercontent.com/u/1820240/supportpages/LDPCpublic.html.